



Jukka Raappana

WINDOWS PHONE -SOVELLUKSEN KEHITYSTYÖ JA JULKAISU

WINDOWS PHONE -SOVELLUKSEN KEHITYSTYÖ JA JULKAISU

Jukka Raappana
Opinnäytetyö
Syksy 2014
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Tekijä: Jukka Raappana
Opinnäytetyön nimi: Windows Phone -sovelluksen kehitystyö ja julkaisu
Työn ohjaaja: Lasse Haverinen
Työn valmistumislukukausi ja -vuosi: Syyslukukausi 2014
Sivumäärä: 69 + 3 liitettä

Työn aiheena oli kehittää ja julkaista älypuhelinsovellus Windows Phone -alustalle sekä tarkastella kehitystyön aikaisia haasteita. Esimerkkisovelluksena kehitettiin strategiapeli, Mylly.

Työn suorittaminen jakaantui pääosin kolmeen vaiheeseen. Ensimmäisessä vaiheessa hahmoteltiin sovelluksen käyttöliittymä ja sen oleelliset perusominaisuudet siten, että pelilaudalle voitiin asettaa pelinappuloita sattumanvaraisesti. Toinen vaihe keskittyi mietintään, jossa pelinappuloiden siirtelyn kannalta huomioitiin muun muassa sääntöjen vaikutus. Lisäksi kehitettiin tärkeä elementti, puhelimen siirrot päättävä peliäly. Lopuksi keskityttiin sovelluksen julkaisuun sekä sovellus lokalisoitiin kolmelle kielelle.

Työn tuloksena syntyi julkaisukelpoinen peli, jolla peruspelaaminen luonnistuu. Jatkokehittävää vielä jää peliälyn kehittäminen seuraavalle vaativammalle tasolle, jonka voittaminen olisi jo vaikeaa.

Asiasanat: mobiililaitteet, mobiilisovellukset, pelit

ABSTRACT

Oulu University of Applied Sciences
Information technology

Author: Jukka Raappana
Title of thesis: Developing and publishing a Windows Phone application
Supervisor: Lasse Haverinen
Term and year when the thesis was submitted: Autumn 2014
Pages: 69 + 3 appendices

The purpose of this thesis work was to develop and publish a smart phone application for Windows Phone, and study challenges which are met during the design work. As an example a strategy game, Mills, was developed.

Work was divided into three main phases. First one concentrated on sketching graphical user interface including its essential features, and be able to just place pieces on the game board in random order. Target of second phase was to take game rules into account, and develop an intelligence, which takes care of calculating how the phone moves pieces. Last phase was to study publishing process and finally publish the application localized into three languages.

As a result the first version of the game application is available in the store. More skilled game level is pending in future release.

Keywords: smartphones, mobile applications, mobile games

ALKULAUSE

Kiitän opinnäytetyön ohjaajaa Lasse Haverista ohjauksesta ja tuesta. Kiitos myös Tuula Hopeavuorelle kieliasun tarkastamisesta. Toivon tämän raportin lukeneiden saavan vinkkejä omien sovellustensa kehitystyöhön.

Oulussa 30.11.2014

Jukka Raappana

SISÄLLYS

| | |
|---|----|
| TIIVISTELMÄ | 3 |
| ABSTRACT | 4 |
| ALKULAUSE..... | 5 |
| SISÄLLYS..... | 6 |
| SANASTO..... | 9 |
| 1 JOHDANTO..... | 11 |
| 2 WINDOWS PHONE..... | 12 |
| 2.1 Historia..... | 12 |
| 2.2 Kehitystyökalut..... | 13 |
| 2.3 Visuaalisuus..... | 13 |
| 2.3.1 Silverlight | 13 |
| 2.3.2 XNA..... | 14 |
| 2.4 Vaatimuksia kehitysympäristölle..... | 14 |
| 2.4.1 Emulaattori..... | 14 |
| 2.4.2 Aidon laitteen käyttö testauksessa..... | 15 |
| 2.5 Ohjelmointikieli..... | 16 |
| 2.6 Koodin kääntäminen | 16 |
| 2.7 WP7 ja WP8..... | 17 |
| 3 KÄYTTÖLIITTYMÄ..... | 18 |
| 3.1 Yleistä kehitystyöstä..... | 19 |
| 3.2 Emulaattorin ja todellisen laitteen erot..... | 19 |
| 3.3 Sivut ja niiden ulkoasu..... | 19 |
| 3.3.1 Näytön asemoinnin vaihtoehtoja..... | 20 |
| 3.3.2 Pelisovelluksen layout..... | 20 |
| 3.3.3 Alasivut..... | 21 |
| 3.3.4 Sovelluspalkki..... | 21 |
| 3.4 Grafiikka ja kuvien piirto..... | 22 |
| 3.5 Näppäimet..... | 23 |
| 3.5.1 Vakionäppäimet..... | 23 |
| 3.5.2 Näppäimet kuvassa..... | 23 |
| 3.6 Kielivalinnat ja kulttuuri..... | 24 |

| | |
|--|----|
| 3.6.1 Globalisointi..... | 24 |
| 3.6.2 Lokalisointi..... | 24 |
| 3.6.3 Kulttuuri..... | 25 |
| 3.6.4 Sovelluksen nimi..... | 25 |
| 3.7 Kyvykkyydet..... | 27 |
| 4 MYLLY-SOVELLUS..... | 28 |
| 4.1 Yleistä..... | 28 |
| 4.2 Puhelimen sovellusvalikko..... | 28 |
| 4.3 Pääsivu..... | 29 |
| 4.4 Valinnat-sivu..... | 31 |
| 4.5 Ohje-sivu..... | 32 |
| 4.6 Tilastot-sivu..... | 34 |
| 4.7 Lokalisointi..... | 34 |
| 4.8 Pelilauta..... | 35 |
| 4.8.1 Ruudukko..... | 35 |
| 4.8.2 Mallinnus koodissa..... | 36 |
| 4.8.3 Indeksien laskenta..... | 37 |
| 4.8.4 Pelilaudan käsitteitä | 38 |
| 4.8.5 Kosketuspisteen muunnos pelipisteeksi..... | 39 |
| 4.9 Pelilaudan luokkia..... | 39 |
| 4.9.1 Pelinappula..... | 39 |
| 4.9.2 Pelipiste..... | 40 |
| 4.9.3 Pelilauta..... | 40 |
| 4.10 Ohjelman runko..... | 40 |
| 4.11 Pelitilan ajastin..... | 41 |
| 4.12 Pelitilan vaiheet..... | 42 |
| 4.12.1 Nappulan lento..... | 43 |
| 4.12.2 Odotus..... | 44 |
| 4.12.3 Idle..... | 44 |
| 4.12.4 Tarkista miilut..... | 44 |
| 4.12.5 Nappulan poisto..... | 44 |
| 4.12.6 Tarkista pelin tilanne..... | 45 |
| 4.12.7 Muut tilat..... | 45 |
| 4.13 Puhelimen siirrot..... | 45 |

| | |
|---|----|
| 4.13.1 Asetteluvaihe..... | 45 |
| 4.13.2 Pelivaihe..... | 46 |
| 4.13.3 Lopetusvaihe..... | 49 |
| 4.14 Pelaajan siirrot..... | 49 |
| 4.14.1 Asetteluvaihe..... | 49 |
| 4.14.2 Pelivaihe..... | 49 |
| 4.14.3 Lopetusvaihe..... | 50 |
| 4.15 Miilut..... | 50 |
| 4.16 Navigointi..... | 51 |
| 4.17 Koodin laajuus..... | 52 |
| 5 SOVELLUKSEN JULKAISU..... | 53 |
| 5.1 Sovelluksen sertifiointi..... | 53 |
| 5.1.1 Sisältö ja sen rajoitukset..... | 53 |
| 5.1.2 Sovelluspolitiikka..... | 53 |
| 5.1.3 Asennuspaketti..... | 54 |
| 5.1.4 Vaatimuksia koodille..... | 54 |
| 5.1.5 Lokalisointi..... | 54 |
| 5.1.6 Ikonit..... | 54 |
| 5.1.7 Esimerkkikuvat sovelluksesta..... | 55 |
| 5.2 Kaupan testaussovellus..... | 55 |
| 5.3 Sovelluksen kaupallisuus..... | 57 |
| 5.4 Kokemukset Mylly-sovelluksen julkaisusta..... | 57 |
| 5.4.1 Automaattitestit..... | 57 |
| 5.4.2 Manuaaliset testit..... | 58 |
| 5.4.3 Sovelluksen lataus kauppaan..... | 59 |
| 5.4.4 Ikäsuositus..... | 60 |
| 5.4.5 Mainostekstit..... | 61 |
| 5.4.6 Palaute..... | 61 |
| 5.4.7 Tilastotietoa..... | 61 |
| 6 POHDINTA..... | 63 |
| LÄHTEET..... | 65 |
| LIITTEET..... | 69 |

SANASTO

| | |
|------------|---|
| DreamSpark | Microsoftin ohjelma, joka tukee teknisiä opintoja ja opetusta tarjoamalla työkaluohjelmistoille ilmaislisenssin |
| DLL | Dynamic Linked Library, dynaamisesti linkitettävä kirjasto, jossa erityistä ohjelmakoodia |
| ESRB | Entertainment Software Rating Board; ikäsuositukset, Yhdysvallat |
| Gt | Gigatavu |
| GUI | Graphical User Interface, graafinen käyttöliittymä |
| HW | Hardware, yleisnimitys laitteen elektroniikalle |
| IDE | Integrated Development Environment, integroitu kehitysympäristö |
| ISS | Isolated Storage, WP:n Flash-muisti, joka on varattu vain sovelluksen käyttöön mm. tiedostojen tallennusta varten |
| Landscape | Yleisnimitys sivusta, jonka vaakasuunta on pitempi kuin pystysuunta, engl. landscape = maisemakuva |
| MUI | Multilingual User Interface, lokalisoinnin yhteydessä käytetty DLL-tiedosto |
| NOP | No Operation, eräänlainen tyhjä toiminto tai käsky |
| OS | Operating System, käyttöjärjestelmä |
| PC | Yleisnimitys tietokoneelle |
| PEGI | Pan European Game Information; ikäsuositukset, Eurooppa |
| Portrait | Yleisnimitys sivusta, jonka pystysuunta pitempi kuin vaakasuunta, engl. portrait = muotokuva |
| RAM | Random Access Memory, laitteen keskus- tai käyttömuisti |
| SDK | Software Development Kit, ohjelmiston kehitystyökalu(t) |

| | |
|----------|---|
| SW | Software, yleisnimitys ohjelmistoille |
| USK | Unterhaltungssoftware Selbstkontrolle, ikäsuositukset/Saksa |
| VS | Visual Studio, Microsoftin ohjelmisto sovelluskehitykseen |
| WWW | World Wide Web, yksi internetin palvelujärjestelmä |
| WP | Windows Phone, yleisnimitys puhelimesta Windows-alustalla. WP8 viittaa käyttöjärjestelmän versioon 8 ja WP7 vanhempaan 7.x |
| WP Store | WP-sovelluskauppa, täältä on mahdollista ladata sovelluksia |

1 JOHDANTO

Älypuhelimien ja muiden mobiililaitteiden käyttäjäkunta on kasvanut viime vuosina voimakkaasti. Sellainen löytyy Tilastokeskuksen mukaan joka toiselta suomalaiselta yli 16-vuotiaalta, ja todennäköisesti osuus olisi yli kaksi kolmannesta, jos nuoremmat ala- ja yläkoululaisetkin laskettaisiin mukaan. (1.)

Jotta käyttäjä saa mobiililaitteestaan enemmän irti, tarvitaan hyviä sovelluksia. Tässä opinnäytetyössä tarkastellaan Windows Phone -sovelluksen kehitystyötä. Esimerkkiprojektina kehitetään peli, mobiiliversio perinteisestä Mylly-lautapelistä. Tarkastelun pääpaino on suunnattu sovelluksen kehitystyön eri vaiheisiin ja sen haasteisiin päättyen sovelluksen julkaisuun, jolloin se on yleisesti saatavilla. Sovelluksen kaupallistamisnäkökohdat rajataan työn ulkopuolelle. Työn aihe on kirjoittajan oma.

2 WINDOWS PHONE

2.1 Historia

Microsoftin kehittämän älypuhelimien Windows Phone -käyttöjärjestelmän ensimmäinen julkaisu tapahtui vuonna 2010. Pääuutuus tuolloin oli uusi moderni Metro-käyttöliittymä kosketusnäytölle. Usein puhuttaessa Windows Phonesta tarkoitetaan yleensä laitetta, jossa on WP-käyttöjärjestelmä. Valmistajia ovat mm. HTC, Samsung ja Microsoft (entinen Nokia). (2.)

WP:n markkinasegmentti on ollut alusta lähtien enemmän kuluttajat kuin yritys-käyttäjät. Tälle markkina-alueelle tulivat mm. Nokiankin ensimmäiset Lumia-puhelimet kuten 800 tai 900, joiden käyttöjärjestelmänä oli aluksi WP7.5 (Mango). Käyttöjärjestelmä päivittyi myöhemmin WP7.8:ksi, joka peri hiukan ominaisuuksia uudemmasta WP8:sta. Tällä saatiin näiden laitteiden käyttöikää pidennettyä varsinkin, kun alkuperäinen ajatus WP8:n käytöstä näissä kariutui laitetason puutteisiin. Taulukossa 1 on esitetty lyhyt vertailu ominaisuuksista. (2.)

Vuonna 2012 julkaistu käyttöjärjestelmän uudempi versio WP8 sai äskettäin päivitysversion. WP8.1 onkin se, millä uusimmat laitteet tänä päivänä toimitetaan. Laitetasolla WP8:ssa on aiempiin nähden mm. selvästi parempi tuki isommille näyttöresoluutioille ja moniydinprosessoreille. Kaikki parannukset olivatkin tarpeen, jotta välimatkaa kilpailijoihin saatiin kurottua pienemmäksi. (2.)

TAULUKKO 1. Nokian Lumia 8xx -sarjan puhelimien perusominaisuuksien vertailu. (3.)

| Malli | Vuosi | Proessori | OS | RAM [Gt] | Resoluutio | Näytön koko |
|-------|-------|------------------|-----|----------|------------|----------------|
| 800 | 2011 | 1,4 GHz, 1x ydin | 7,8 | 0,5 | 800 x 480 | 3,7" (9,4 cm) |
| 820 | 2013 | 1,5 GHz, 2x ydin | 8,1 | 1 | 800 x 480 | 4,3" (10,9 cm) |
| 830 | 2014 | 1,2 GHz, 4x ydin | 8,1 | 1 | 1280 x 720 | 5,0" (12,7 cm) |

2.2 Kehitystyökalut

Microsoftin tarjoamat sovelluskehitystyökalut (engl. software development kit, SDK) ovat kehittäjille ilmaisia, riittää kunhan rekisteröi asennuksen. Sen sijaan sovellusten julkaisusta ja jakelusta joutuu maksamaan kertaluontoisen rekisteröintimaksun, joka on noin 15 € yksityishenkilöille ja 75 € yrityksille. Microsoft on juuri äskettäin (syyskuu 2014) julkistanut uuden kehittäjäohjelman, joka ei enää sisällä vuosimaksuja. (4.)

DreamSpark-ohjelman mukaisesti opiskelijoille sovellusten julkaisu on ilmaista. Tosin julkaisujen määrä on tällöin rajoitettu. (5.)

Kehitysympäristönä on Microsoftin Visual Studio Express tai vaihtoehtoisesti Expression Blend. Nämä ovat ns. integroituja ohjelmistonkehitysympäristöjä (engl. integrated development environment, IDE), jotka sisältävät koodieditorin, graafisen käyttöliittymän suunnittelutyökalun, virheenkorjaustyökalun sekä emulaattorin. (6.)

2.3 Visuaalisuus

Expression Blend on osittain Visual Studion (VS) laajennus ja se sisältää monipuolisemman graafisen kehitystyökalun vaikkapa animaatioiden tekemiseen. Käyttöliittymän visuaalinen suunnittelu on tärkeä osa nykyaikaisen sovelluksen kehityskaarta. Parhaimmillaan se erotetaan varsinaisesta koodaustyöstä niin, että graafinen suunnittelija suunnittelee käyttöliittymän ja ohjelmoija keskittyy varsinaiseen koodiin. (6.)

2.3.1 Silverlight

WP-sovellusten ulkoasun kehittämisen yhteydessä puhutaan usein Silverlightista. Se on eräänlainen kehys, joka tarjoaa helppokäyttöiset käyttöliittymäkomponentit sovelluksen hyödynnettäväksi. Komponentit ovat tietyn tyylin mukaisia, siispä niiden ulkoasu, käyttäytyminen ja fontit mukailevat tätä samaa tyyliä. Näin erilaisten sovellusten välillä säilyy tietty yhtenevä tuntuma. (7.)

Silverlightin toimintamalli on tapahtumakeskeinen. Tämä tarkoittaa sitä, että aina jonkin tapahtuman (engl. event) seurauksena kutsutaan sen käsittelevää metodia. Tyypillisiä tapahtumia ovat

- näppäimen painallus
- kuvan kosketus
- tekstikentän muokkauksen aloitus
- tekstikentän muokkauksen lopetus
- sivun lataus.

2.3.2 XNA

XNA-kehys puolestaan on käytössä usein vaativimmissa pelisovelluksissa, koska se tarjoaa paremman ja välittömän graafisen rajapinnan. Varsinkin 3D-pelimaailmaan XNA on omiaan. Nämä kaksi kehystä on mahdollista yhdistääkin samaan sovellukseen, mutta se vaatii kehittäjiltä enemmän osaamista. (8.)

2.4 Vaatimuksia kehitysympäristölle

Työkaluista on saatavilla eri versiota. Ne kulkevat osittain käsi kädessä käyttöjärjestelmän versioiden kanssa, kuten taulukko 2 esittää.

TAULUKKO 2. Visual Studio Express versioiden eroja. (9.)

| Versio | Tuetut käyttöjärjestelmät | Vaatimuksia PC:lle |
|--------|---------------------------|--------------------|
| 2010 | WP7.1 | Windows 7 |
| 2012 | WP7.x, WP8 | Windows 8 Pro 64x |

2.4.1 Emulaattori

Usein raskain osa kehitysympäristöä on emulaattori. Se siis jäljittelee PC:ssä oikeaa puhelinta, jossa sovellusta voi kokeilla ja testata.

Varsinkin WP8:n emulaattori asettaa suuria vaatimuksia PC:lle. Tarvitaan siis PC, jonka käyttöjärjestelmänä on Windows 8.1 Pro 64x sekä koneen emolevyllä pitää olla tuki Hyper-V-virtualisoinnille. Käytännössä se tarkoittaa mahdollisuutta ajaa emulaattoria ns. virtuaalikoneena. Emolevyllä Hyper-V on mahdollista, jos ne tukevat jompaakumpaa seuraavista:

- Intelin prosessori, SLAT (Second Level Address Translation)
- AMD:n prosessori, RVI (Rapid Virtualization Indexing). (9.)

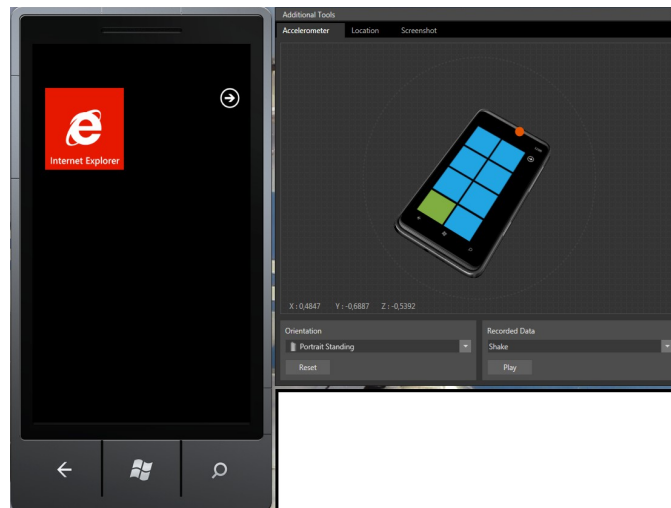
Sovelluksen testaus

Emulaattoria käytettäessä sovelluksen testaus ja virheenpaikannus eli debugaus on samantyylistä kuin PC-sovelluksenkin kanssa. Koodia voi askeltaa, siihen voi asettaa pysäytyskohtia ja tarvittaessa vaikka muuttaa muuttujan arvoa lennosta. Sovelluskoodin kaatuessa emulaattori usein myös osaa kertoa hieman lisätietoa, missä meni pieleen ja miksi.

Koodiin voi upottaa myös diagnostiikkaa, esimerkiksi tekstitulostuksia on mahdollista tulostaa VS:n ikkunaan samaan tapaan kuin kyseessä olisi PC:n komentoikkuna ja print-käskyt. Parhaimmillaan tällä voi seurata sovelluksen ajoa ja vaikkapa haluttujen muuttujien arvoja.

Puhelimen näytön kosketuksia jäljitellään hiiren avulla, mutta näppäimistöltä voi tarvittaessa kirjoittaa etenkin tekstiä suoraan. Lisäksi puhelimen perusominaisuudet ovat ainakin virtuaalisesti läsnä. Esimerkiksi voi

- muuttaa GPS:n paikkatietoa kartalla
- jäljitellä puhelun soittamista tai tekstiviestin lähetystä
- muuttaa puhelimen asentoa kädessä (kuva 1).



KUVA 1. Emulaattori ja sen lisätyökaluikkuna avattuna.

2.4.2 Aidon laitteen käyttö testauksessa

Kun sovellus toimii riittävän hyvin emulaattorin kanssa, voi sovelluksen asentaa laitteeseen kokeilua varten. Jotta tämä onnistuu, laite pitää rekisteröidä kehittä-

jän laitteeksi. Kun sovellus on asennettu laitteeseen, sitä voi käyttää kuten mitä tahansa muutakin asennettua sovellusta. Ainoa ero aidosti julkaistuun sovellukseen on, että näytön sivussa on pieniä debuggauksesta kertovia numeroita aivan kuten emulaattorin näytölläkin. Lisäksi rajoituksena on se, että laitteeseen voi asentaa korkeintaan kaksi kokeiltavaa sovellusta.

Debuggaus

Debuggaus laitteen kanssa on hiukan rajoitetumpaa, ehkä hitaampaakin. Mikäli haluaa jotain lisätietoa sovelluksen suorituksesta ajonaikana, sellaiset ominaisuudet on lisättävä koodiin ja tulostettava väliaikaisesti näytölle.

Käytettäessä vanhempaa WP7-laitetta tarvitaan vielä ns. Zune-sovellus USB-yhteyden muodostamiseksi laitteen ja PC:n välille. Puolestaan WP8-laite pystyy muodostamaan USB-yhteyden ilman erillistä yhteyssoftaa. Huomattava on kuitenkin, että WP8-laitteen käyttö tarvitsee vähintään VS2012:n ja Windows 8:n. (10.)

2.5 Ohjelmointikieli

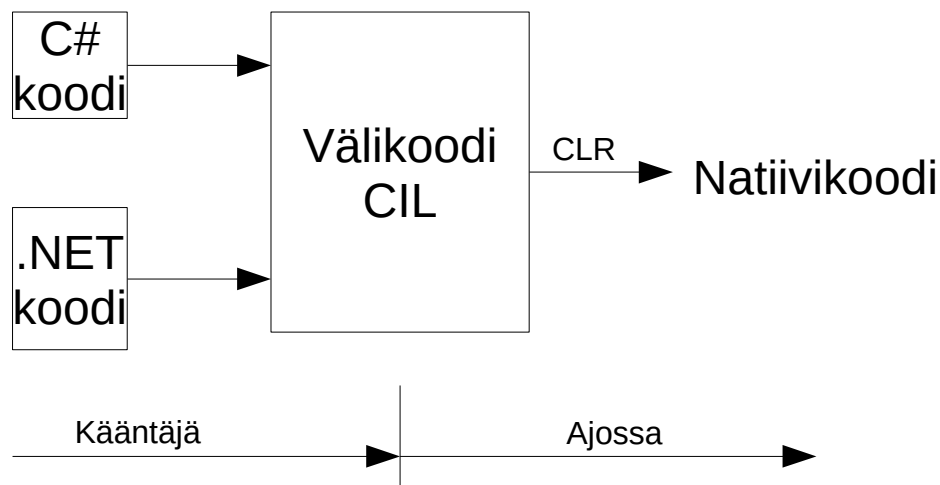
WP:n sovellusten kehittämiseen käytetään yleensä Microsoftin kehittämää C#-ohjelmointikieltä (engl. C sharp). Se on oliopohjainen ja muistuttaa sekä Javaa että C++-kieltä. Kielen kehitystyön päätavoitteena onkin ollut yhdistää Javan helppous ja C++:n tehokkuus. C#:n ohella käytetään XAML-merkkauškieltä, jolla kuvataan Silverlightin käyttöliittymäkomponentit. VS:n projektissa yhtä sivua varten on varsinainen C#-kooditiedosto **xxx.CS** sekä ulkoasutiedosto **xxx.XML**. (11.)

2.6 Koodin kääntäminen

Sovelluksen ajossa WP hyödyntää Microsoftin .NET-virtuaalikonetekniikkaa. Käytännössä tämä tarkoittaa sitä, että sovelluksen koodia ei käännetä ohjelmointityökalulla valmiiksi ohjelmatiedostoksi (kuten .exe) vaan välikoodiksi (engl. common intermediate language, CIL). Kuten kuva 2 havainnollistaa, vasta käynnistettäessä sovellus ajoon käyttöjärjestelmän ajonaikainen moottori (engl. common language runtime, CLR) tekee välikoodista ajokelpoista ns. natiivia koodia. (12.)

Tällä käännöstavalla saavutetaan mm. seuraavia etuja:

- tuki poikkeuskäsittelylle (muistin ylivuoto yms.)
- muistin varaus ja vapautus
- roskien keruu
- säikeet.



KUVA 2. Käännöksen vaiheistus. (12.)

2.7 WP7 ja WP8

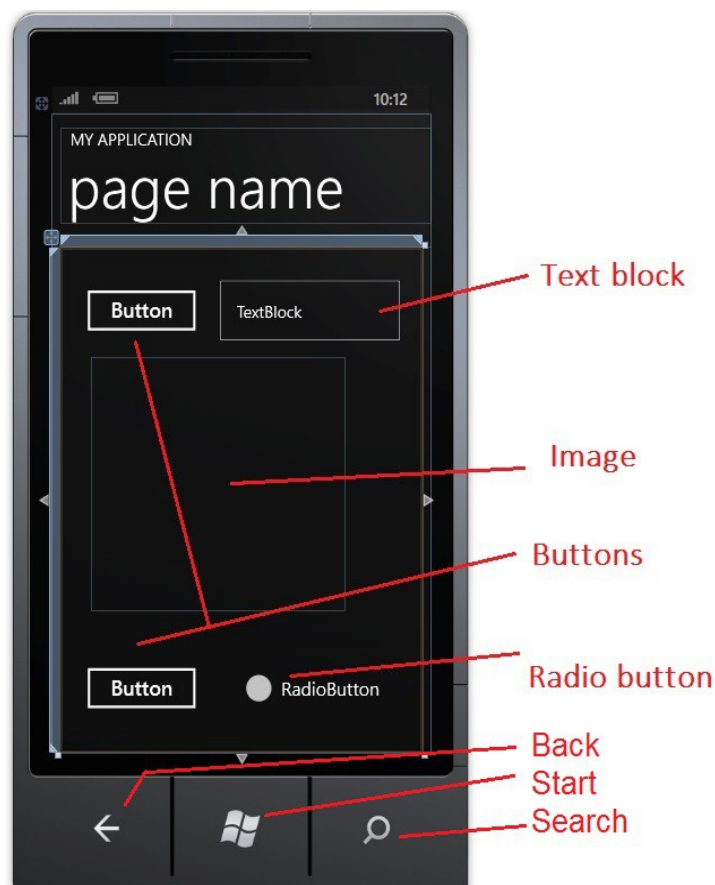
Useimmiten WP7:lle kehitettyjä sovelluksia on mahdollista käyttää WP8:lla ilman muutoksia koodiin tai sen uutta käännöstä. Toiminnallisuus on kuitenkin testattava WP8-emulaattorissa tai itse laitteessa, jotta saadaan käytännön varmuus. Vasta käännettäessä WP7:lle kehitettyä koodia aidoksi WP8-sovelukseksi, on tarkistettava tiettyjä perusasioita. Tästä taulukossa 3 on muutama esimerkki. (13; 14.)

TAULUKKO 3. Esimerkkejä WP7:n ja WP8:n koodieroista. (13.)

| Tapaus | WP7.1 | WP8 |
|-----------------------------------|--|--|
| ISS.FileExist | Jos annettu tiedoston nimi null, palauttaa false-arvon | Menee poikkeuskäsittelyyn |
| Kulttuuri | Vaihto vaikuttaa kaikkiin sovelluksiin | Vaihto vaikuttaa vain sen hetkiseen säikeeseen |
| BeginRead EndRead tai write | Input ja output käsitellään synkronisesti | Input ja output käsitellään asynkronisesti |

3 KÄYTTÖLIITTYMÄ

Kun sovellus käynnistetään, käyttäjälle avautuvaa näkymää kutsutaan käyttöliittymäksi (engl. graphical user interface, GUI). Näytöllä kerrallaan oleva näkymä voidaan ajatella sivuksi. Työkalu tarjoaa valmiina erilaisia kontrollielementtejä, joita voidaan asettaa sivulle sovelluksen koodin hyödynnettäväksi. Kuvassa 3 on muutamia esimerkkejä niistä.



KUVA 3. Tyypillisiä WP-sovelluksen kontrollielementtejä (Silverlight) sekä WP:n omat oletusnäppäimet. (15.)

WP:ssa on vain muutama oletusnäppäin. Näytön alalaidassa ovat näppäimet Back (takaisin), Start (aloitus) ja Search (Bing-haku). Puhelimen sivulla ovat näppäimet äänen voimakkuuden säätöön, kameran laukaisin sekä virtakytkin. Yleensä perussovelluksessa näistä tarvitaan vain Back-näppäintä liikkumaan sivulta takaisin tai sulkemaan sovellus. (15.)

3.1 Yleistä kehitystyöstä

Sovelluksen kehitystyö on paras tehdä järkipäisesti tyvestä puuhun eli keskittyä aluksi perustoiminnallisuuteen. Ensin on tärkeämpää, että jokin toiminto toimii, eikä esimerkiksi se, että se on toteutettu mahdollisimman hienosti tai parhaalla tavalla. Muutoin on vaarana, että kehitystyön aikaa kuluu liikaa epäolennaiseen. Toki koodauksessa on pidettävä mielessä koodin sujuvat muutosmahdollisuudet, jotta jälkeinpäin tapahtuvat parannukset tai optimoinnit ovat toteutettavissa järkevästi.

Esimerkkinä pelissäni pääsivulla on ylhäällä näppäimet, keskellä pelilauta ja alimpana informaatiokenttä tekstile. Näidenkin asemointi toisiinsa nähden voi olla parempi jotenkin toisin, mutta alussa sen pohtimiseen ei kannata käyttää liikaa aikaa. Samoin kannattaa menetellä värien valinnassa. Alkuvaiheessa valittuja värejä on mahdollista muuttaa hyvin helposti myöhemmin.

3.2 Emulaattorin ja todellisen laitteen erot

Yksi olennainen ero on hyvä ymmärtää alusta pitäen sovelluksen testauksessa. Ohjelmointityökalun emulaattorissa tietyt asiat ovat erilaisia kuin varsinaisessa laitteessa. Emulaattorissa painalluksiin käytetään hiiren kursoria, puhelimen ruudulla osoittimena on sormi. Näiden välinen kokoero on valtava. Tässäkin tapauksessa näytön toimintoja ei kannata miettiä liian hienoksi testaamatta ideaa ensin varsinaisessa laitteessa. Omassa pelissäni huomasin tämän mm. pelinappulan koossa. Kokeilujen perusteella muutin pelinappulan halkaisijaa puolitoistakertaiseksi alkuperäiseen nähden.

3.3 Sivut ja niiden ulkoasu

Kun tekijällä on tiedossa, millaista sovellusta hän alkaa kehittämään, kehitystyö oikeastaan aloitetaan hahmottelemalla sovelluksen pääsivun ja mahdollisten alisivujen ulkoasu (engl. layout), eli siis mitä noilla sivulla on ja minkälaista esitystapaa käytetään. Myös sivujen määrä olisi hyvä tietää alussa ainakin suurin piirtein, mutta niiden lisääminen jälkeinpäin on myös helppoa.

3.3.1 Näytön asemoinnin vaihtoehtoja

Sivujen layoutin suunnittelussa ensimmäinen määräävä tekijä on se, millä tavalla laitetta haluttaisiin pideltävän kädessä sovellusta käytettäessä. Näytettävä sivu voi olla siis pysty- tai vaakasuuntainen, tai molempia, jolloin laitteen näyttö kääntyy automaattisesti seuraamaan käyttäjän sen hetkistä laitteen pitotapaa käsissään. Tämä antaa omat mahdollisuutensa, mutta voi olla myös hankaloitava seikka. Mm. Smashing Magazinen artikkelissa (16) kerrotaan muutamien mallikuvien kera hyviä vinkkejä, kuinka tätä voi hyödyntää. Yksinkertaisimmillaan sovellus vain kääntyy ja näytettävä informaatio rajautuu sen mukaan, mutta miinuspuolena tässä on tietysti se, että kerrallaan näkyvän informaation määrä usein vähenee oltaessa vaakänäytössä. Parempi malli lienee se, jossa näkyvillä oleva informaation määrä pysyy samana, vain sen paikka ruudulla mukautuu valittuun pitotapaan. Oma viehätöksensä on ideoissa, joissa senhetkisen sivun informaatio tarkentuu uudeksi alasivuksi käännön myötä. Taulukossa 4 on kuvattu mahdollisia asemointeja.

TAULUKKO 4. Näytön asemointeja ja niiden eroja.

| Moodi | Kuvaus | Huomaa |
|-----------|--|--|
| Portrait | Sivu pystysuorassa | Automaattinen kääntö mahdollista |
| Landscape | Sivu vaakasuorassa | |
| Pivot | Pystysuorassa oleva sivu, jatkuu alaspäin ja oikealle | Sisältö optimoitu sivun kokoon |
| Panoraama | Pystysuorassa näytössä oleva isohko vaakasuorainen panoraamasivu, joka näkyy vain osittain näytöllä kerrallaan, jatkuu alaspäin ja sivulle | Sisältö optimoitu panoraaman kokoon, joka on suurempi kuin näytön koko |

3.3.2 Pelisovelluksen layout

Pelisovelluksessa on käytännössä usein valittava joko pysty- tai vaakasuuntainen näyttötapa. Jos peli käyttää ohjailuun suoraan puhelimen asentoantureita, automaattista näytön kääntöä ei pystytä hyödyntämään, koska se sekoittaisi

ohjauksen. Lisäksi näissä tapauksissa laitetta ohjaillaan ja käytetään pääsääntöisesti kahdella kädellä pelikonsolityyliin, luonnollisin vaihtoehto on vaakasuuntainen näyttö.

Pystysuuntainen näyttötapa on usein käytännöllinen sovelluksissa, jossa ohjailu tapahtuu perinteisemmin näytön yksittäisten kosketusten kautta. Tällöin laitetta pidetään toisessa kädessä, ja vapaan käden sormin annetaan syötteitä näytön kautta sovellukselle. Tässä tapauksessa luonnollisin pitotapa laitteelle on kaapeampi sivu sormien välissä, mutta poikkeuksiakin varmasti löytyy. Pelissäni päädyin pystysuuntaiseen näyttötapaan.

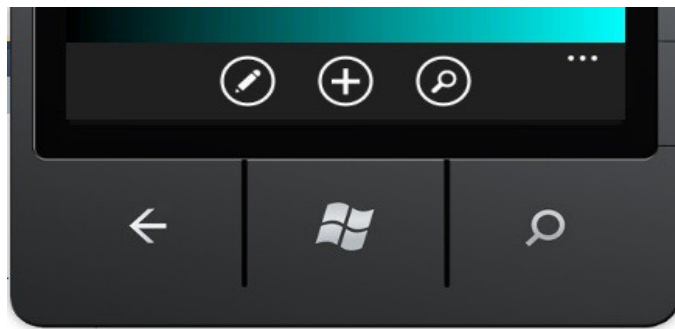
3.3.3 Alasivut

Sovelluksen ominaisuudet kannattaa jakaa useammalle sivulle. WP:n perusresoluutio on 800 x 480 pikseliä, mikä voi vaikuttaa isolta, mutta se on oikeastaan yllättävän pieni sovelluksen näkökulmasta. Alasivuille saakin helposti ryhmitellä ehkäpä vähemmän käytettäviä ominaisuuksia kuten pelin asetukset tai ohjeet. Sivusta saa myös helposti tavanomaista isomman käyttämällä pystysuunnassa vieritettävää sivua. Mikään ei tietenkään estä tekemästä pääsivustakaan vierityskelpoista, mutta omasta mielestäni pääsivu tulisi pitää mahdollisimman selkeänä.

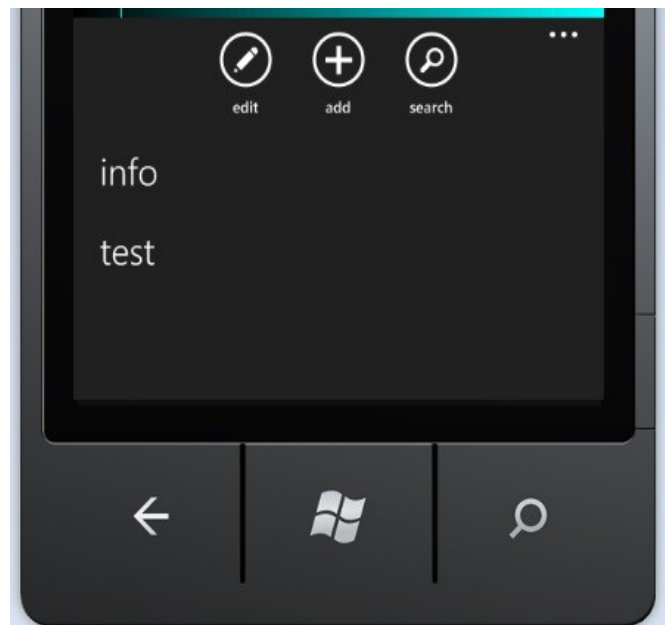
3.3.4 Sovelluspalkki

Sovelluspalkin (engl. application bar) avulla on mahdollista helposti luoda ikään kuin Windows-ohjelman menuvalikko; tosin se on WP-puhelimessa sivun alalaidassa. Se voi pitää sisällään valmiita kuvakkeita eli ikoneita, jotka toimivat näppäinten tavoin, tai avautuvan pop up -tyylisen menun, jonka rivit toimivat eräntäisina hyperlinkkeinä.

Ikonien etuna on pieni koko, joten ne vievät näytöltä vain vähän tilaa. Niistä saa parhaassa tapauksessa kuvaavia, jolloin ikonin tarjoama toiminto ymmärretään helposti. Miinuspuolena on rajoitettu määrä, sillä yhdelle sivulle on mahdollista laittaa vain neljä ikonia. Palkki ei myöskään käänny vaakatasoon sivun mukana. Menuvalikon etuna on pieni koko minimoituna, ja puolestaan avattuna se voi täyttää vaikkapa koko näytön. Toki rivien määrä on tällöin rajaava tekijä. Kuvis- sa 4 ja 5 on esimerkkejä. Pelissäni menuvalikko on käytössä ohjesivulla.



KUVA 4. Tyypillinen sovelluspalkki ja sen ikonit.



KUVA 5. Edellisen sovelluspalkin menuvalikko avattuna.

3.4 Grafiikka ja kuvien piirto

Pelisovelluksen graafinen hahmottelu on tarkkaa puuhaa. Vaikka kyseessä olisi pelkkä pelilaudan tapainen kuva, sen geometria on syytä miettiä matemaattisesti, jotta koodissa ei jouduta mahdottoman eteen. Käytännössä kaupallisissa sovelluksissa grafiikan hahmottelee ammattilainen, graafinen suunnittelija. Tällä taataan visualisesti paras ja jopa ehkä taiteellinenkin näkökulma. Aloitteijan on syytä pitäytyä toimivuudessa ja ehkäpä hänen taiteelliset lahjansa määräävät lopputuloksen.

3.5 Näppäimet

Sivuilla tarvitaan erilaisia painonäppäimiä (engl. button), joilla käynnistetään haluttu operaatio kuten uusi peli tai siirrytään alasivulle. Helpoin ratkaisu on käyttää valmista näppäin-kontrollia. Tällöin sen kaikki perusominaisuudet ovat helposti hyödynnettävissä sovelluksessa, ja ohjelmoitavien näppäimien käyttö on suoraviivaista.

3.5.1 Vakionäppäimet

Riippuen sovelluksen tilasta, kuten onko peli käynnissä (aloitettu) vai ei, tai onko peli päättynyt, tarvitaan vain tiettyjä näppäimiä. Parhaiten tämä onnistuu hyödyntämällä ohjelmoitavia eli ns. soft-key-näppäimiä, joilla ohjataan sovelluksen käyttöä kulloiseenkin tilanteeseen sopivimmaksi. Esimerkiksi *Uusi peli* -näppäin on tarpeeton, kun peli on jo käynnissä. Sen sijaan voidaan aktivoida vaikka *Undo*-näppäin.

Näppäimiä voidaan myös sammuttaa ja sytyttää, siis tehdä näkymättömäksi tai näkyväksi Visibility-attribuutin avulla. Mikäli yhdellä näppäimellä on vain kaksi tai maksimissaan kolme eri toimintoa, on ehkä helpointa käyttää soft-key-näppäiminä päällekkäin olevia erillisiä näppäimiä, joista vain yksi on kulloinkin näkyvissä. Näppäimen painallus kutsuu sen omaa painallukseen linkitettyä tapahtumankäsittelijää, jolloin kullekin näppäimelle on oma itsenäinen metodinsa.

Yleiskäyttöisemmässä koodissa on vain yksi näppäin, jonka parametreja ohjataan sovelluksen tilan mukaan. Tällöin tapahtumankäsittelijässä on tunnistettava kutsuva olio, jotta voidaan suorittaa oikea osa koodista. Toki tarvittaessa käsittelijääkin voidaan dynaamisesti vaihtaa, jolloin ehkä ollaan lähellä aiempaa päällekkäisten näppäinten tapausta.

3.5.2 Näppäimet kuvassa

Pelisovelluksessa näytöllä on useimmiten iso grafiikkakuva, joka mallintaa pelialuetta tai -lautaa. Tällöin voi olla epäkäytännöllistä käyttää valmista, usein isohkoakin näppäin-oliota. Itse kuvaan voidaan tehdä näppäintä muistuttavia kohtia, joiden painallukset on suodatettava varsinaisen muun pelialueen koskeuksista. Vaihtoehtoinen ja ehkä helpompi tapa on käyttää kuvaa kuvassa, eli

pelialueen kuvan päällä on toinen kuva itse näppäimestä. Käyttöjärjestelmä osaa tässä tapauksessa luonnollisesti kutsua päällimmäisenä olevan kuvan metodeja esim. sitä kosketettaessa.

3.6 Kielivalinnat ja kulttuuri

3.6.1 Globalisointi

Globalisointi tarkoittaa sovelluksen mukautumista mobiililaitteelle asetetun käyttöalueen mukaan. Tyypillisimpiä käyttäjälle näkyviä asioita ovat mm. lukujen desimaalierotin (pilkku vai piste), päiväyksen näyttötapa ja oletusvaluutta. Tällä ominaisuudella pyritään saamaan sovellus toimimaan ja näyttäytymään käyttäjälle ennestään tutulla tavalla. (17.)

3.6.2 Lokalisointi

Lokalisointi puolestaan tarkoittaa sovelluksen mukautumista mobiililaitteelle asetetun käyttökielen mukaan. Sen sijaan, että esimerkiksi näppäinten teksti asetettaisiin suoraan koodissa, käytetään viittauksia resurssitiedostoon hiukan muuttajanimien tapaan. Alla esimerkkinä tekstin asetus suoraan sekä resurssitiedoston avulla:

- `nappulanTeksti = "Uusi peli"` (tekstin suora asetus)
- `nappulanTeksti = AppResources.NewGameText.`

Yleensä tapana on, että viitataan kokonaisiin lauseisiin yksittäisten sanojen sijasta. Tämä varmistaa varsinkin monimutkaisimmissa tapauksissa hyvän kieliopin käännöksessä. (17.)

Resurssitiedostoja on sovelluksella niin monta kuin on tuettuja kieliäkin. Yksi niistä toimii aina ns. neutraalina kielenä eli on oletuskieli, mikäli laitteen kielivalinnalle ei löydy sovelluksessa vastinetta. Yleensä neutraalina kielenä on englanti. Taulukossa 5 on esitetty esimerkkinä muutama poiminta resurssitiedostoihin asetetuista viittaus-tekstipareista.

TAULUKKO 5. Esimerkki resurssitiedostojen viittaus-tekstipareista.

| Viittauksen nimi | Teksti, neutraali | Teksti, FI | Teksti, SE |
|------------------|-------------------|------------|------------|
| MainPageText | myMills | Mylly | Kvarn |
| NewGameText | New game | Uusi peli | Nytt spel |
| HelpPageText | Help | Ohje | Hjälp |

Lokalisointia varten projektiin lisätään luokka **LocalizedStrings.cs**, jonka kautta viittaukset kielen mukaiseen tekstiin käytännössä suoritetaan koodissa.

Tässä yhteydessä erikoistapauksena voidaan mainita sovelluspalkki, joka yleensä on määritelty XAML-koodissa. Jostain syystä dynaaminen tiedonsi-donta ei onnistu tässä tapauksessa suoraan, vaan se on tehtävä kiertotietä. Sovelluspalkki on luotava C#-koodissa dynaamisesti, jolloin myös dynaamiset viittaukset resurssitiedostoon toimivat. (18.)

3.6.3 Kulttuuri

Kulttuurin voidaan ajatella tarkoittavan hiukan samaa asiaa kuin PC:ssä maa-koodin. Kulttuurin valintaan on WP:ssä myös omat koodinsa, kuten Suomen koodi fi-FI. (17.)

VS:n projektissa asetetaan sovelluksen tukemat kulttuurit. Projektin määritys-tiedostossa (XML-muodossa .csproj) on määritys esimerkiksi:

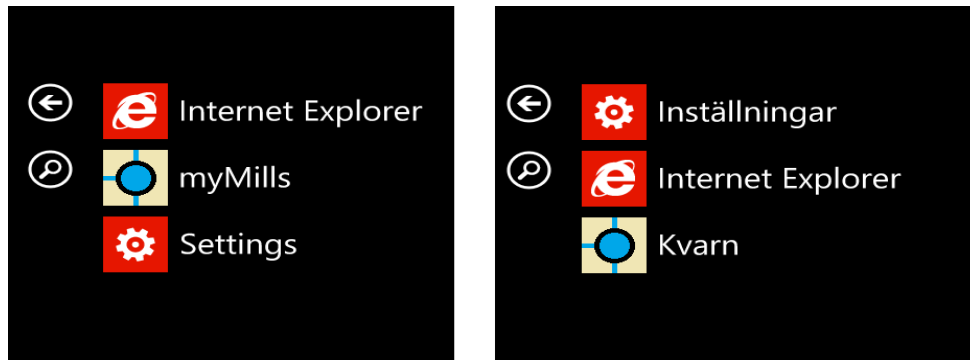
`<SupportedCultures>us-US;fi-FI;sv-SE;</SupportedCultures>`. Tässä tapauk- sessa kielet ovat siis englanti (US), suomi ja ruotsi. Neutraalikielen valinta löytyy puolestaan tiedostosta **AssemblyInfo.cs**. Yleensä nämä voidaan asettaa pro- jektin ominaisuuksien kautta, mutta myös tiedostojen suora muokkauskin on joissain tapauksissa ainoa vaihtoehto.

Projektiin lisättävät resurssitiedostot on nimettävä yhtenevästi kulttuurin kanssa. Mm. suomenkielinen tiedosto on **AppResources.fi-FI.resx**.

3.6.4 Sovelluksen nimi

Sovelluksen nimen lokalisointi käynnistysvalikossa on tehtävä hiukan eri tavalla. Sama pätee ns. tiili-ikoneihin (engl. tile), jotka ovat WP:n etusivulla näkyvät

pikakäynnistyskuvakkeet. Tässä luvussa sitä on kuvattu lyhyesti. Esimerkkikuvat ovat kuvassa 6. (19.)



KUVA 6. Sovelluksen nimen lokalisointi käynnistysvalikossa, esimerkkinä ruotsin- ja englanninkieliset käynnistysvalikot.

Lokalisointia varten luodaan erityiset DLL-resurssitiedostot, joiden kautta nimet asetetaan. Nämä yleensä tehdään Visual Studio C++ (ei siis VS WP Expressillä) -työkalulla, jolla voidaan kääntää Win32-projekti. On hyvä huomata, että sovellushan itsessään ei tarvitse näitä, koska se ei edes ole käynnissä siinä vaiheessa, kun nimiä näytetään. Win32-projektiin lisätään resurssina merkkijonotaulukko, johon tulee vain kaksi taulukossa 6 mainittua muuttujaa. (19.)

Yleensä ikonissa ja käynnistysvalikossa nimet ovat selkeyden vuoksi samat, mutta näin ei tarvitse olla. Varsinkin pitkä sovellusnimi voi olla järkevä lyhentää, koska tiili-ikonin tila on rajallinen.

TAULUKKO 6. Esimerkki, DLL-tiedostojen string-muuttujat. (19.)

| Muuttuja | ID | Arvo |
|--------------|-----|--------------------------------------|
| AppTitle | 100 | Mylly (näkyvä käynnistysvalikossa) |
| AppTileTitle | 200 | Mylly (näkyvä isossa tiili-ikonissa) |

Resurssitiedostoon viitataan projektin **WMAppManifest.xml** määrittelyssä, ja siellä on esimerkiksi mainittu `Title="@AppResLib.dll,-100"`. Resurssitiedosto tarvitaan jokaista tuettua kulttuuria varten, ja muut kuin neutraalikieli nimetään

ns. MUI-tiedostoksi (engl. multilingual user interface, MUI), jossa on mukana kulttuuriin viittaava koodi. VS-projektiin lisättävät DLL-resurssitiedostot ovat

- neutraali; **AppResLib.dll**
- suomi; **AppResLib.dll.040b.mui** (projektissa content)
- ruotsi; **AppResLib.dll.041d.mui** (content).

3.7 Kyvykkyydet

Mobiililaitteella on erilaisia kyvykkyyksiä (engl. capabilities), joilla tarkoitetaan laitteen tiettyjen erikoisominaisuuksien käyttöä. Tällaisia ovat mm.

- verkkoyhteys
- paikannus
- mikrofoni
- kamera
- www-selain
- sensorit
- käyttäjän todennuspalvelut
- peliserveripalvelut
- kalenteri, puhelinluettelo.

Sovelluskaupassa on mainittava, mitä kyvykkyyksiä sovellus hyödyntää. Kun sovellusta asennetaan laitteeseen, niin samassa yhteydessä käyttäjä antaa luvan sovellukselle käyttää mainittuja ominaisuuksia. Tyypiesimerkki voisi olla verkkoyhteyden käyttö, mikä tarkoittaa käyttäjälle mahdollisia datansiirtomaksuja. Kyvykkyydet määritellään projektin tiedostossa **WMAppManifest.xml**.
(20.)

4 MYLLY-SOVELLUS

Tässä luvussa kuvataan tarkemmin esimerkkipelin toteutusta. Kuvauksessa on keskitytty olennaisimpiin osiin, joten ihan kaikkea sovelluksen piirteitä ei ole esitetty.

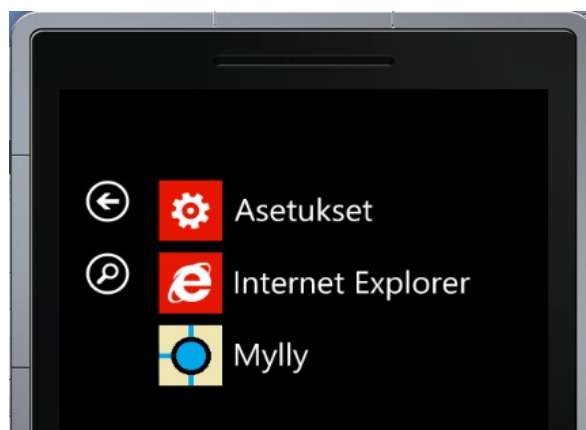
4.1 Yleistä

Mylly-sovellus on yhteensopiva vanhemman WP7.1:n kanssa. Yksi syy tähän on ollut kevyempi kehitysympäristö. Kyvykkyyksistä toistaiseksi sovellus hyödyntää www-selainta sekä verkkoyhteyttä. Näistä jälkimmäistä ei itse asiassa käytetä, mutta todennäköisesti se tulee selaimen mukana oletuksena.

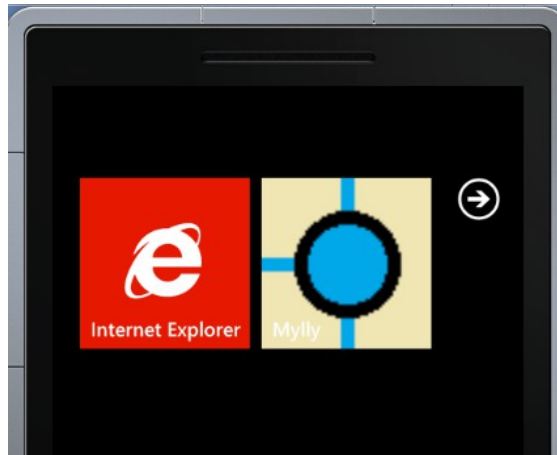
4.2 Puhelimen sovellusvalikko

Kun sovellus asennetaan puhelimeen, ilmestyy sovellusvalikkoon sen oma ikoni kuten kuva 7 osoittaa. Se on helppo suunnitella kehitystyön aikana millä tahansa piirtotyökalulla ja se tallennetaan projektiin useimmiten nimellä

ApplicationIcon.png. Kuvassa 8 on esimerkki vastaavasti puhelimen etusivulla näytettävästä ns. tiili-ikonista. Sen piirtämisessä voi mukailla samaa teemaa kuin pienemmässä ikonissakin on. Tämä kuva tallennetaan projektiin nimellä **ApplicationTile.png**. Luvussa 5.1.6 on selostettu kuvien kokoon ja formaattiin liittyvät vaatimukset tarkemmin.



KUVA 7. Sovellus-ikoni.



KUVA 8. Tiili-ikoni.

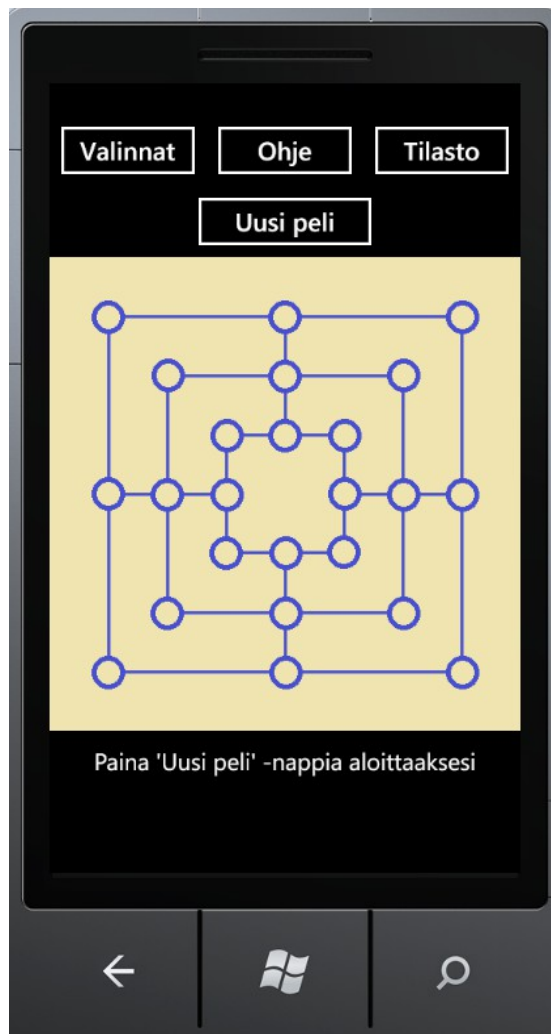
Sovellukselle on mahdollista asettaa ns. genre eli sovelluksen laji. VS:n projektissa on manifesti-tiedosto (**WMAppManifest.xml**), jossa tämä asetus on. Yleensä oletuksena on yleissovellus, mutta muuttamalla se peliksi, saadaan peli näkymään puhelimen pelinäppäimen takaa avautuvassa valikossa. Perusvaihtoehdot ovat

- Genre = apps.normal, yleissovellus
- Genre = apps.games, pelisovellus (Mylly on tätä lajia). (21.)

4.3 Pääsivu

Kun Mylly-sovellus avataan, aukeaa kuvan 9 mukainen näkymä. Ylhäällä on näppäimet, joilla päästään alisivuille, sekä näppäin uuden pelin aloitukseen. Näppäimet on toteutettu Silverlightin tarjoamina vakionäppäiminä. Keskellä on kuva pelilaudasta ja sen alla tekstilohko, johon tulostetaan tarpeellista infoa käyttäjälle.

Back-näppäimellä peli sulkeutuu suoraan, jos peli ei ole kesken. Muussa tapauksessa kysytään vahvistus sovelluksen sulkemiselle. Toistaiseksi kesken olevaa peliä ei tallenneta siten, että uusi käynnistys palaisi tähän tilanteeseen takaisin.



KUVA 9. Pääsivu.

Logo

Useimmat sovellukset käynnistyessään näyttävät ensimmäiseksi logon. Se on kaupallisissa sovelluksissa paikallaan ja eräänlainen tavaramerkki. Periaatteessa se on helppoa esimerkiksi hyödyntäen WP:n ominaisuutta, joka sovelluksen käynnistymisen aikana näyttää kuvaa (**SplashScreenImage.jpg**). Myllysovellus toteuttaa logon juuri näin. (22.)

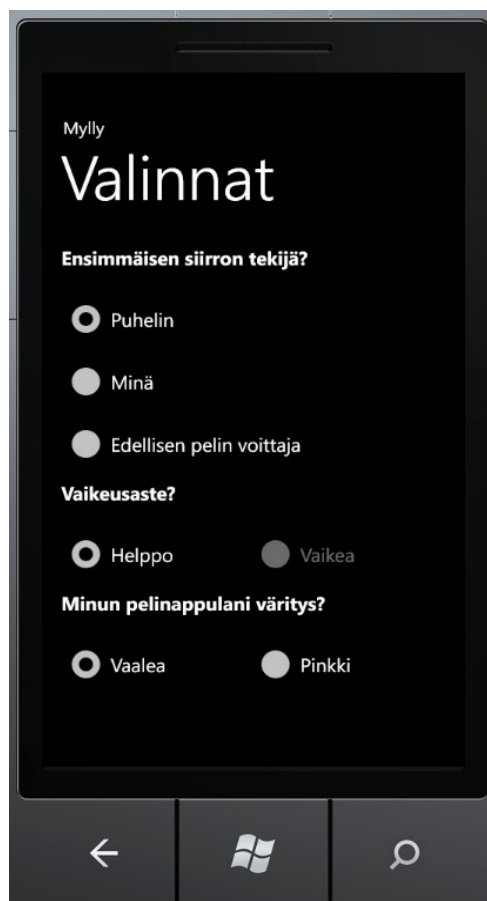
WP8-sovellusten käynnistysajat ovat jo sen verran nopeita, että edellä kuvattu tapa ei enää välttämättä ole suositeltavin, mikäli haluaa logon näkyvän riittävän kauan. Silloin joutuu lisäämään aikaa keinotekoisesti tai ehkä kannattavinta on ohjelmoida koodiin oma metodi, joka näyttää logoa vaikkapa animoitujen tervetuloivotuksien kera.

4.4 Valinnat-sivu

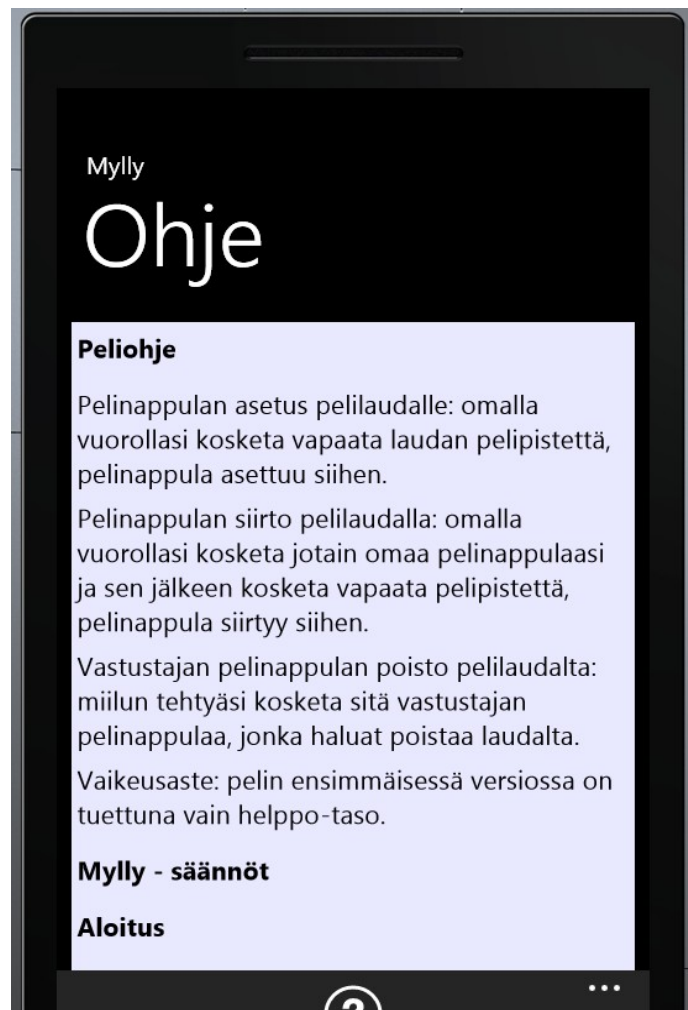
Valinnat-sivun näppäintä painamalla päästään muokkaamaan muutamia pelaajan kannalta tarpeellisia asetuksia. Pelaaja voi valita

- pelin aloittajan
- pelin vaikeusasteen
- oman nappulan värityksen.

Pelaajan valinnat talletetaan tiedostoon (ISS), joten ne säilyvät eri pelikertojen välillä. Periaatteessa valintoja voisi hyvinkin monipuolistaa. Esimerkiksi nappulan väritys tai ihan kuvakin olisi mahdollista tuoda valittavaksi useammasta vaihtoehdosta. Vaikeusasteista *vaikea* on harmaana, koska se ei ole tuettuna ensimmäisessä versiossa. Kuvassa 10 on sivu avattuna. Back-näppäimellä päästään takaisin pääsivulle.



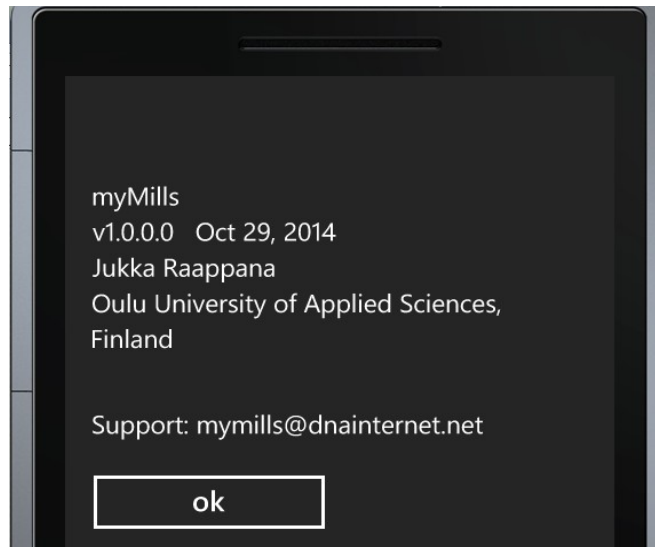
KUVA 10. Valinnat-sivu avattuna.



KUVA 11. Ohje-sivu avattuna.

4.5 Ohje-sivu

Ohje-näppäintä painamalla avautuu kuvan 11 mukainen uusi sivu, johon tulostuu pelin säännöt ja muita ohjeita. Sivun alalaidassa olevassa sovelluspalkissa on *tietoja*-toiminto, joka tulostaa ilmoitusikkunaan tietoja sovelluksen versiosta (kuva 12).



KUVA 12. Ohje-sivun sovelluspalkin ikonin painallus avaa ilmoitusikkunan.

Varsinainen ohjeen tulostava osio on toteutettu mielenkiintoisella tavalla. Se ei ole mikään tekstiä suoraan näyttävä rakenne vaan aito www-selain. Sovellus-paketissa on mukana ohje HTML-muodossa, joka tarvittaessa kopioidaan sovel-luksen ISS-muistiin, josta selain sen avaa. Myös kuvat tallennetaan samaan paikkaan.

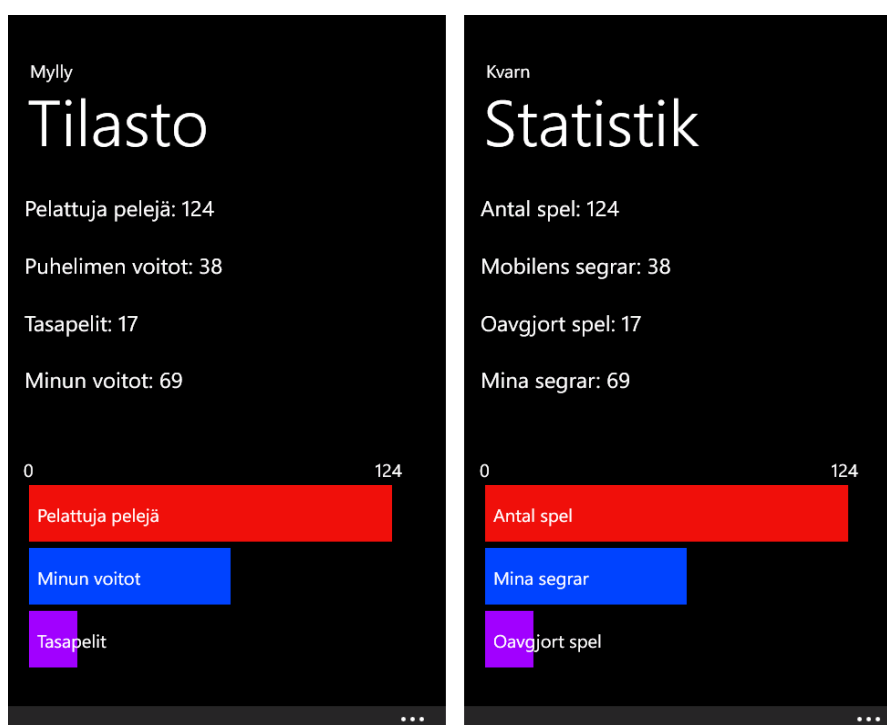
Tällä ratkaisulla on useampiakin etuja. Paras niistä on se, että ohjeeseen voi liittää helposti kuvia. Lisäksi selain tarjoaa sivun vierityksen ja zoomauksen, joten isonkin teksti- ja kuvamäärän näyttö onnistuu. Puhelimessa on myös yleinen asetus selaimen tekstin koolle, joten se käyttää käyttäjän asettamaa tekstikokoa luonnostaan.

Koska kyseessä on selain, luonnollisesti tiedosto voisi sijaita jollain serverillä ja se ladattaisiin verkon yli. Tällöin mahdolliset puutteet olisi helppo korjata ilman sovelluksen raskasta päivitysprosessia kauppaan. Samoin tarjontaa erikielisille ohjeille saataisiin sujuvasti lisää. Nyt tuettuna oleville kielille on kullekin oma tiedostonsa, kuvia tarvitaan vain yhdet. Puhelimen Back-näppäimellä päästään takaisin pääsivulle.

4.6 Tilastot-sivu

Tilastot-sivun avaamalla nähdään pelaamisesta kerättyä tilastotietoa. Se on myös esitetty graafisina palkkeina (kuva 13).

Sivun alaosassa on sovelluspalkki, jonka ainoana toimintona on mahdollisuus nollata tilasto. Tässä käytössä on vielä minimoitu palkki, joka vie mahdollisimman vähän tilaa varsinaisen näytön alueelta. Näkyvissä on siis vain kolme pistettä, ja niitä koskettamalla palkki laajenee ja teksti *nollaa tilasto* on näkyvisä. Sitä näpäyttämällä avautuu ilmoitusikkuna, jossa pyydetään vahvistamaan nollaustoiminto. Puhelimen Back-näppäimen kautta päästään pääsivulle.



KUVA 13. Tilastot-sivusta ovat tässä esimerkkinä ruotsin ja suomen kielelle lokalisoidut versiot.

4.7 Lokalisointi

Mylly-sovellus tukee kolmea lokalisoitua kulttuuria, jotka ovat

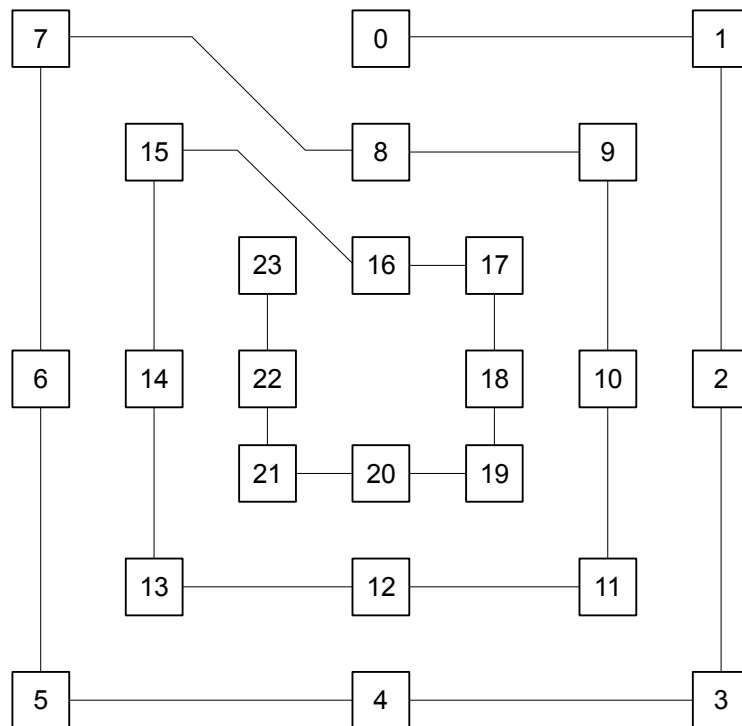
- englanti (oletuskieli, us-US)
- suomi (fi-FI)
- ruotsi (sv-SE).

dinaatiston origo. Tätä siirtymää pitää tarpeen mukaan joko lisätä tai vähentää laskennassa, jotta eri tilanteissa saadaan oikea arvo.

4.8.2 Mallinnus koodissa

Pelilaudan toteutuksessa yksi olennaisimmista seikoista on, miten sen toiminnot sulautuvat koodiin ja ovat helposti käytettävissä erilaisiin tapauksiin.

Pelissäni päädyin numeroimaan pelilaudan pisteet kuvan 15 mukaisesti. Nolla-kohta on keskellä ylhäällä ja numerot kiertävät myötäpäivään. Kun neliöllä on kahdeksan pistettä, keskimäinen neliö alkaa luonnollisesti numerosta kahdeksan, ja sisin numerosta kuusitoista.



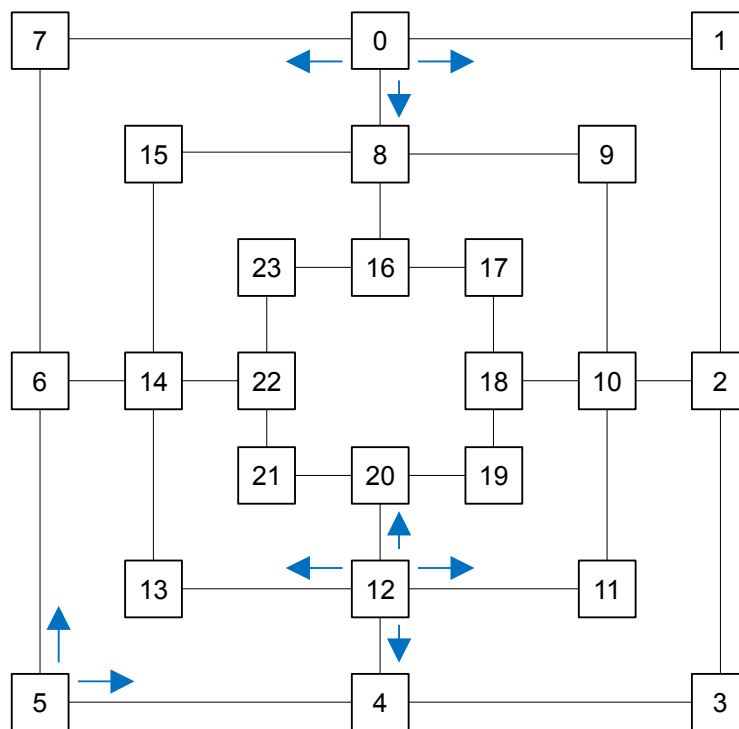
KUVA 15. Pelilaudan pisteiden järjestys.

Koodin kannalta tuosta saadaan helposti muodostettua taulukko, jossa on 24 alkia. Pelissäni alkioden tyyppi määräytyy luokan **PieceSpot** mukaan; tästä jäljempänä enemmän.

Numerointitavasta saadaan mm. seuraavanlaisia etuja:

- siirryttäessä samalla neliöllä indeksi joko kasvaa tai pienenee
 - erikoistapaus mm. 0->7
- jos indeksi on pariton, tällöin ei neliötä voi vaihtaa
 - siirtyminen onnistuu vain eteen tai taaksepäin
- jos indeksi on parillinen, siirtyminen neliöltä toiselle on mahdollista
 - indeksi joko kasvaa tai pienenee kahdeksalla (± 8)
 - uloin ja sisin neliö sisältää vain kolme suuntaa.

Kuvassa 16 on havainnollistettu siirtymistä eri tapauksissa.



KUVA 16. Pelilaudalla siirtymisen eri tapauksia.

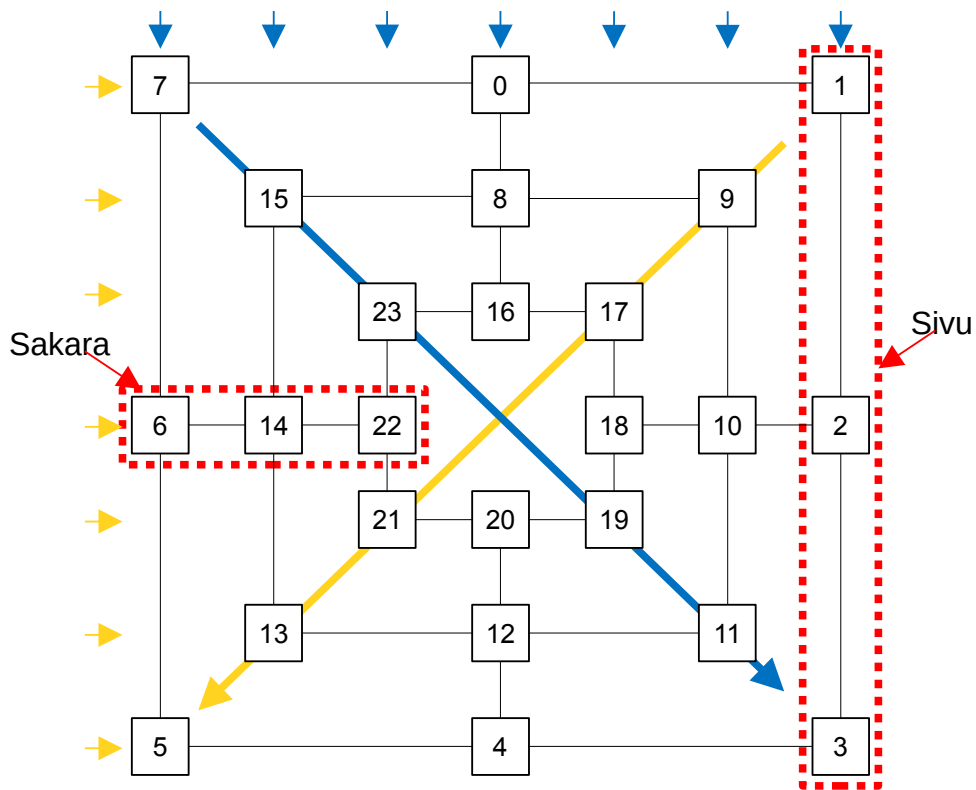
4.8.3 Indeksien laskenta

Indeksin laskennassa on helppo hyödyntää mm. jakojäännös-operaattoria. Esimerkkejä näistä metodeista on esitetty liitteessä 1.

4.8.4 Pelilaudan käsitteitä

Kun pelilautaa tarkastelee yksinkertaistaen, voidaan ajatella kunkin neliön sivua (kuvassa 17 esim. 1-2-3) yhtenä tapauksena, ja käytännössä kaikki sivut mallintuvat samalla tavalla. Vastaavasti laudalle muodostuu eräänlainen risti, jonka yksi sakara on esim. kuvassa 17 pisteet 6-14-22.

Pelin koodissa tehdään mm. yhden miilun tarkistus keskipisteiden suhteen, joka askeltaa parillisin numeroin ($N = N + 2$). Näin laskurin avulla voidaan tarkistaa aina yhden sivun pelipisteet $N - 1$, N ja $N + 1$ kerrallaan. Tarkistuksen toisessa vaiheessa tutkitaan vastaavasti neljä ristin sakaraa laskurin arvoilla $N=8,10,14$ ja 16 sekä tällöin ylä- ja alapisteet ovat $N - 8$ ja $N + 8$.



KUVA 17. Pelilaudan käsitteet neliön sivu ja sakara, sekä koordinaattien laskentasuunnat.

4.8.5 Kosketuspisteen muunnos pelipisteeksi

Kun pelilautaa kosketetaan, täytyy kosketuskohdan koordinaatit muuntaa ensin ruudukon mukaiseksi lähimmäksi pelipisteeksi kuten aiemmin mainittiin. Koodissa lasketaan matemaattisesti vastaava lähin pelipiste hyödyntäen seuraavia oletuksia:

- neliöiden pystysuorat sivut (kuvassa 17 pienet tummat nuolet)
- neliöiden vaakasuorat sivut (kuvassa 17 pienet vaaleat nuolet)
- virtuaalinen neliön lävistäjä, kulkusuunta kello neljään (kuvassa 17 paksu tumma nuoli)
- virtuaalinen neliön lävistäjä, kulkusuunta kello kahdeksaan (kuvassa 17 paksu vaalea nuoli).

Koodissa on eri tapauksille oma yhtälö, jossa on alkupiste (X, Y) sekä ns. delta eli muutosarvopari (muutosX, muutosY). Esimerkkinä lävistäjällä oikealta ylhäältä vasemmalle alas olevista ruuduista lasketaan pelipisteen koordinaatti käyttäen alkupisteenä (420, 60) ja delta-arvona (-60, 60). Näin ollen alkupisteestä seuraava piste on siis $(420, 60) + (-60, 60) = (360, 120)$. Erikoiskohdat kuten laudan keskipiste pitää erikseen ohittaa laskennassa.

Laskettujen koordinaattien etäisyyttä kosketuspisteeseen (= Pythagoraan kolmion hypotenuusa) vertaillaan ja ruudukosta valitaan pelipiste lyhimmän hypotenuusan mukaan. Laskentatapa on ehkä hiukan raskas, mutta tehokas ja lähin pelilaudan piste saadaan luotettavasti haettua. Jakojäännös-operaattorilla voisi päättelyn tehdä ehkä kevyemmin, mutta koska pelilauta ei ole vierekkäisten pelipisteiden etäisyyksien kannalta symmetrinen, valitsin esitetyn tavan. Vertailussa etäisyytenä käytetään suoraan saadun hypotenuusan neliötä eikä siitä oteta neliöjuurta, koska se ei vertailun kannalta ole tarpeellista. Koordinaatit muutetaan lopulta indeksiksi.

4.9 Pelilaudan luokkia

4.9.1 Pelinappula

Pelinappula on yksi oleellisimmista luokista. Sen attribuutteja ja metodeja on lueteltu liitteen 3 taulukoissa 8–12.

Pelinappulat luodaan pääsivulla, joten luokan konstruktorille (liite 1) annetaan parametrina näytön canvas eli se olio, jossa nappulan kuva-image halutaan olevan. Useimpia tämän luokan metodeja käytetään paikan tai sen muutoksen laskentaan.

Kun pelinappulan kuvaa näpätetään ruudulla, sille on oma tapahtumankäsittelijänsä. Lisäksi apuna on ulkopuolinen tapahtumankäsittelijä, joka on asetettu pääsivun luokkaan. Sen päätarkoitus on indikoida nappulan liikutteluun tarkoitettujen tapahtumankäsittelijöiden suorituksen loppua. Esimerkiksi kun nappulaa on näpäytetty, asetetaan se läpinäkyväksi osoittamaan käyttäjälle, että juuri se on esivalittu. Tämän jälkeen välittyy tieto valitusta nappulasta pääsivun tapahtumankäsittelijän kautta sen metodien käytettäväksi. Jos kyseessä olisi vaikka nappulan poisto ruudulta, pääsivun metodit voivat asettaa kyseisen nappulan pois näkyvistä.

4.9.2 Pelipiste

Pelipiste-luokassa on tärkeimpänä osoitin **Piece**-luokan olioön sekä mm. laskuri, johon lasketaan, onko kyseinen piste miilussa vai ei. Erikoistapauksena huomioitakoon, että miiluja voi olla yhdellä nappulalla jopa kaksi samaan aikaan. Liitteessä 3 on lueteltu luokan attribuutteja ja metodeja.

4.9.3 Pelilauta

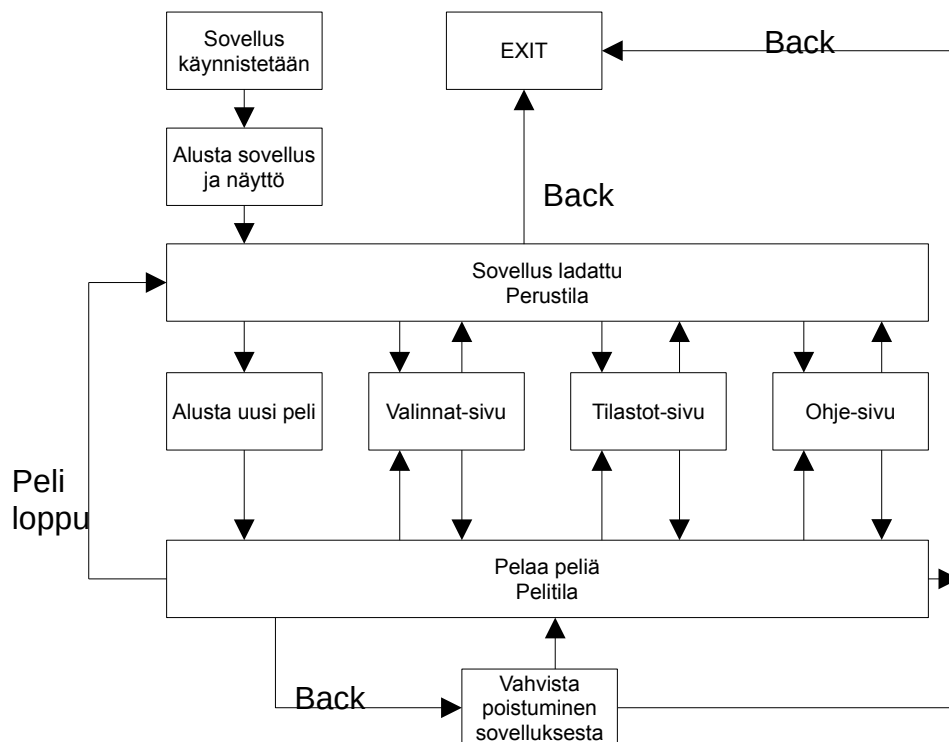
Pelilauta-luokka on oikeastaan vain eräänlainen määritysluokka, jossa on tärkeimmät asiat koottu samaan luokkaan. Tärkeimpänä on pelipisteistä muodostettu staattinen taulukko, johon yksiselitteinen viittaus voidaan tehdä muista luokista. Apuna viittauksessa käytetään metodeja, jotka palauttavat osoitteen taulukoon.

4.10 Ohjelman runko

Kuvassa 18 on kuvattu ohjelman vaiheita. Kun sovellus käynnistetään, ensin alustetaan ja ladataan pääsivu. Tämän jälkeen ohjelma jää odottamaan käyttäjän toimenpiteitä. Käytännössä odotetaan näppäimen painallusta.

Mikäli painetaan *Uusi peli* -näppäintä, pelitila alustetaan ja nollataan tarvittavat attribuutit. Pelitilassa laite ja käyttäjä vuorollaan tekevät siirtoja.

Mikäli painetaan *Valinnat*, *Ohje* tai *Tilastot*, siirrytään sen mukaiselle alisivulle. Alasivulta palataan takaisin Back-näppäimen kautta. Paluu tapahtuu edelliseen tilaan, siis joko perustilaan tai pelitilaan riippuen, kummassa tilassa oltiin ennen alasivulle siirtymistä. Puolestaan Back-näppäimen painallus perustilassa sulkee sovelluksen, mutta pelitilassa kysytään vahvistus poistumiselle.



KUVA 18. Ohjelman tiloja.

4.11 Pelitilan ajastin

Koska WP-sovellus on tapahtumakeskeinen, pelitilassa täytyy käyttää ajastinta, jotta tarvittavia metodeja pystytään kutsumaan. Jos painettiin näppäintä *Uusi peli*, alustetaan taustalle ajastinsäie (engl. thread), joka suorittaa noin 100 ms:n välein ajastintapahtumaa. WP:ssa on tarjolla kaksikin ajastinluokkaa kuten taulukosta 7 nähdään. Mylly-sovellus hyödyntää näistä ensimmäistä.

TAULUKKO 7. Esimerkki ajastinluokista. (22; 23.)

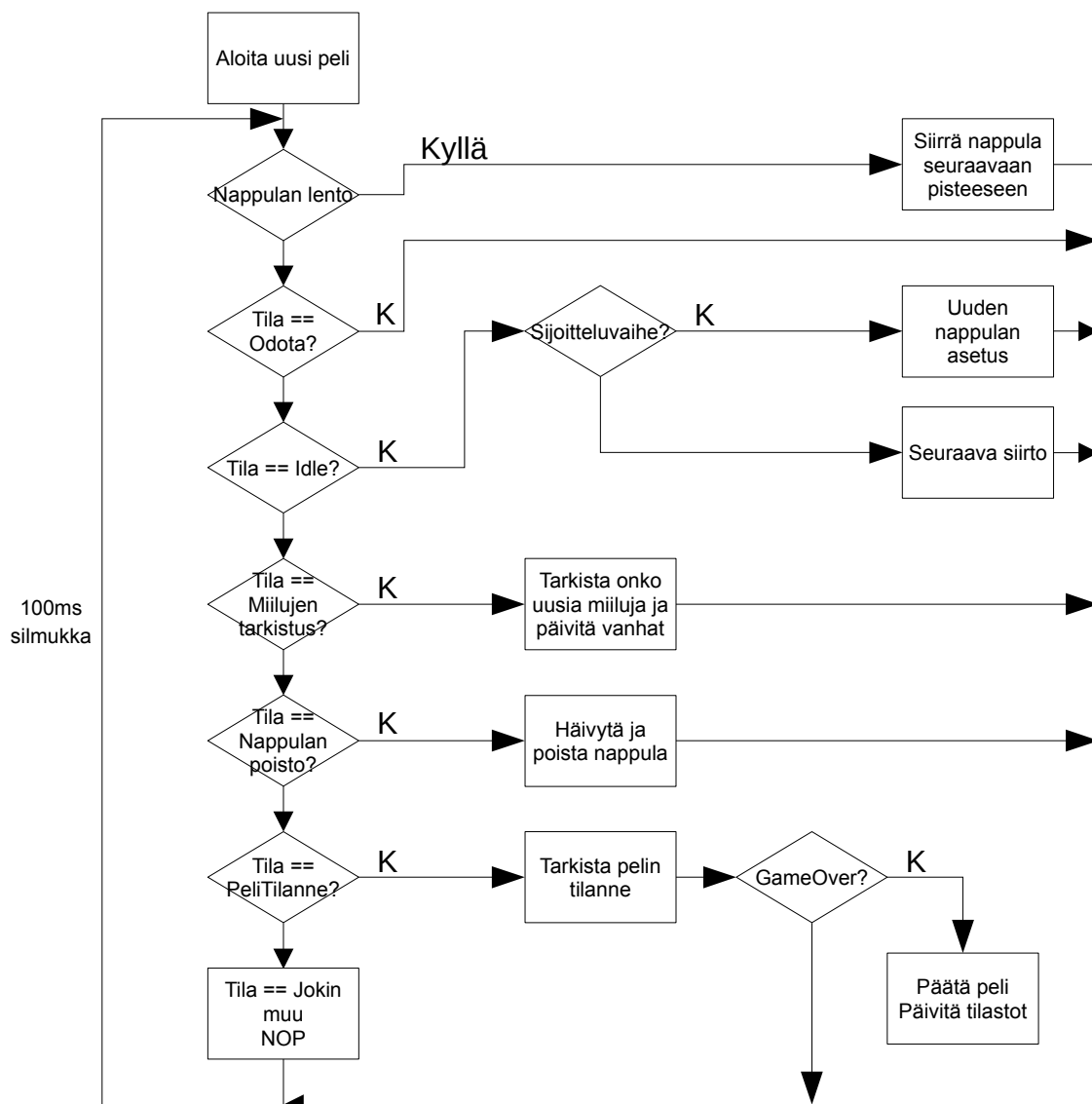
| Ajastinluokka | Kuvaus |
|------------------|---|
| DispatcherTimer | Kevyempi ajastin, jonka tapahtumankäsittelijälle asetetaan aika T [ms], jonka välein se suoritetaan. Start ja stop toiminnot. Huomaa T on minimi odotusaika, ei siis taattu suoritusten tasaväli. |
| BackgroundWorker | Monipuolisempi ajastin. Asetetaan aika T [ms], jonka välein <i>työ</i> suoritetaan. Pystyy raportoimaan <i>työn</i> etenemisestä. Start ja stop toiminto, sekä tukee myös cancel-operaatiota. |

4.12 Pelitilan vaiheet

Koodissa käytetään enum-muuttujaa kuvaamaan pelivaihetta. Sen arvoja ovat

- odotus
- idle
- siirto käynnissä
- siirron tarkastus
- miilujen tarkistus
- pelaajan miilu
- puhelimen miilu
- pelin tarkistus
- nappulan lento
- peli päättynyt.

Kuvassa 19 on pelivaiheiden vuokaavio.



KUVA 19. Ajoitussilmukan vuokaavio.

Seuraavissa alaluvuissa käytetään puhelimen käyttäjästä nimitystä pelaaja.

4.12.1 Nappulan lento

Jos jokin nappula on siirrettävä pelilaudalla kohdasta A pisteeseen B, lasketaan tällöin näiden pisteiden välille tietty suora ja sopivat askellusparametrit, jolla suoraa pitkin pystytään liikkumaan, sekä aktivoidaan lentotila. Jos silmukkaan tultaessa havaitaan jonkin nappulan lentotilan olevan käynnissä, siirretään sitä nappulaa parametrien mukaan. Käyttäjä havaitsee näytöllä tällöin nappulan,

joka lentää pisteestä A pisteeseen B. Lentotila päättyy, kun kohdepiste B on saavutettu.

4.12.2 Odotus

Odotus-tilaa käytetään lisäämään aikaa joissain tapauksissa suorituksen aikana. Muutoin koodin suorituksessa mentäisiin liian nopeasti eteenpäin, eikä pelaaja ehtisi mukaan tai havaita informaatiota tekstikentässä. *Odotus*-tilaa käytetään antamaan näennäinen lisäaika pelin alustuksessa sekä puhelimen siirtojen alussa estämään liian nopea siirto.

4.12.3 Idle

Idle-tila on eräänlainen perustila, josta aina aloitetaan seuraava siirto kutsuamalla siirron alustavaa metodia. Riippuen pelivuorosta, kutsutaan joko puhelimen siirron laskevaa koodia tai alustetaan pelaajan siirto.

Siirtojen tekemisessä on myös kaksi vaihetta:

- nappuloiden asetteluvaihe, siihen asti kun kummankin pelaajan yhdeksän pelinappulaa on saatu asetettua laudalle
- pelivaihe, joka alkaa asetteluvaiheen jälkeen.

4.12.4 Tarkista miilut

Tarkista miilut -tilaan saavutaan aina, kun siirtovuorossa oleva pelaaja on saanut siirron suoritettua. Mikäli uusi miilu on muodostunut, aktivoidaan *nappulan poisto* -koodi miilun tehneelle pelaajalle.

Kun poistettava pelinappula on tiedossa, aktivoidaan *nappulan poisto* -tila.

4.12.5 Nappulan poisto

Nappulan poisto -tilaan saavutaan, kun poistettava pelinappula on tiedossa eikä sitä ole täysin vielä poistettu pelilaudalta. Käytännössä tässä tehdään sen häivytyks hitaasti siten, että pelinappulan läpinäkyvyys-ominaisuutta muutellaan. Pelinappula vilkkuu aluksi kaksi kertaa, jonka jälkeen se häipyä hitaasti näytöltä. Lopulta poistettu pelinappula-olio nollataan näkymättömään perustilaan.

Häivytyksessä hyödynnetään myös *odotus*-tilaa, jolla pidennetään sekvenssiä yli sekuntiin. Muutoin pelaaja ei välttämättä ehtisi havaita, mikä pelinappula katosi näytöltä.

4.12.6 Tarkista pelin tilanne

Pelin tilanne tarkistetaan jokaisen siirron jälkeen. Tarkistus on kaksivaiheinen. Mikäli jokin pelinappula on juuri poistettu laudalta, on tarkistettava jäljelle jääneiden lukumäärät. Lisäksi aina on tarkistettava myös mahdollinen erikoistilanne, jossa seuraavana pelivuoroon tulevan pelaajan pelinappuloiden siirtomahdollisuudet lasketaan. Selvitetään siis, onko vuorossa oleva pelaaja motitettu ilman mahdollisuutta enää siirtää omaa pelinappulaansa.

4.12.7 Muut tilat

Muita tiloja käytetään ohittamaan ajastin, käytännössä siis niissä ei ajastimen kannalta tehdä mitään (tyhjä toiminto eli ns. no operation, NOP). Osa näistä tiloista on sellaisia, että niiden suoritus kestää monikertaisesti yli silmukan toistoajan. Esimerkiksi *siirto käynnissä* -tilan kesto on täysin riippuvainen pelaajan toiminnasta, jos hänellä on siirtovuoro.

4.13 Puhelimen siirrot

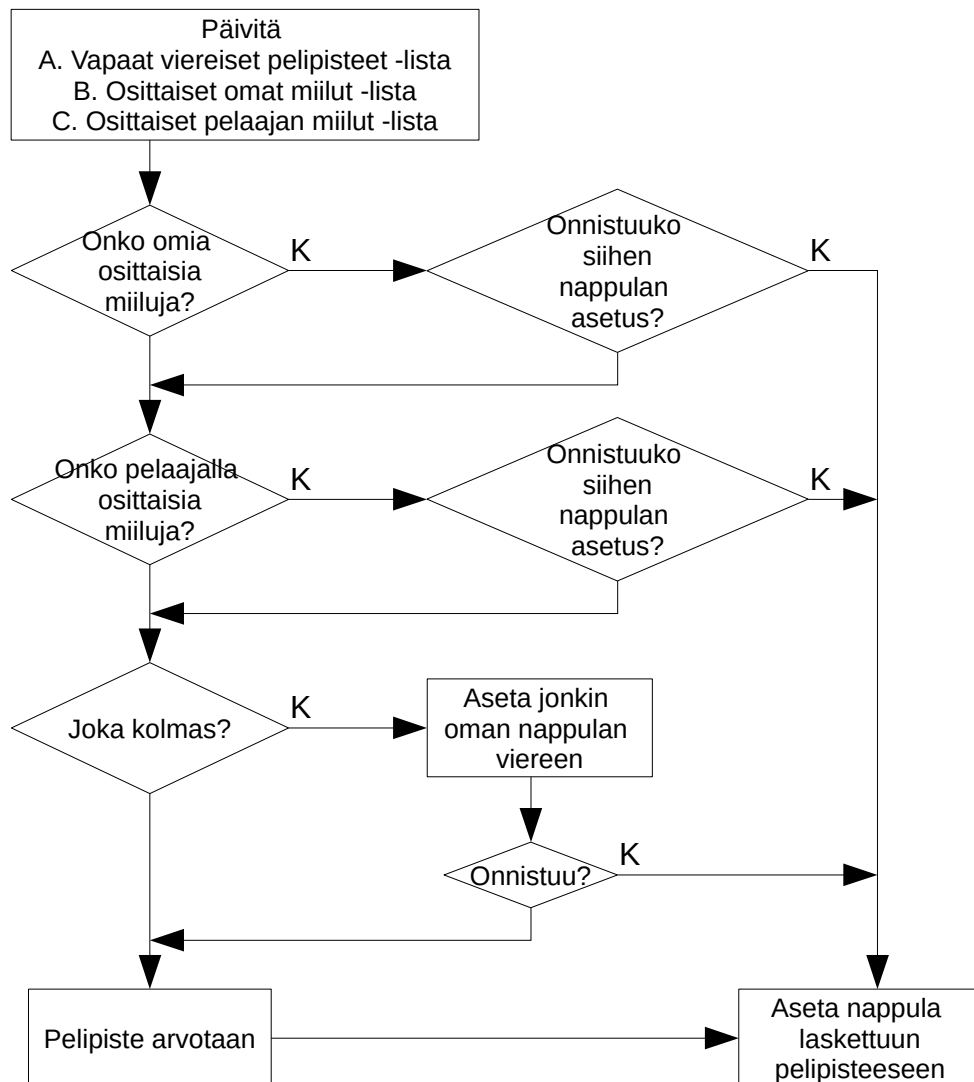
Toistaiseksi pelissä on olemassa vain yksi vaikeusaste (helppo). Tässä luvussa kuvataan vain siihen liittyvä logiikka. Kun siirrettävä pelinappula on tiedossa, sen siirtoon käytetään aina *nappulan lento* -tilaa.

4.13.1 Asetteluvaihe

Asetteluvaiheen aikana puhelimen siirrot tehdään kuvan 20 vuokaavion mukaan. Kaikilla pelinappuloilla on käytössä oma lista, johon voidaan tallettaa vieriset vapaat pelipisteet. Niitä hyödynnetään uuden nappulan asettamisessa.

Samoin pidetään listaa osittaisista miiluista (omat sekä pelaajan), eli sellaisista, joissa on jo kaksi pelinappulaa ja joihin miilun aikaansaamiseksi riittää lisätä enää yksi. Nämä ovat etusijalla nappulan asettamiselle. Muutoin asetetaan

nappula arpomalla tai joka kolmannen nappulan tapauksessa jonkin oman viereen. Tämä useimmiten aikaansaa osittaisen miilun.



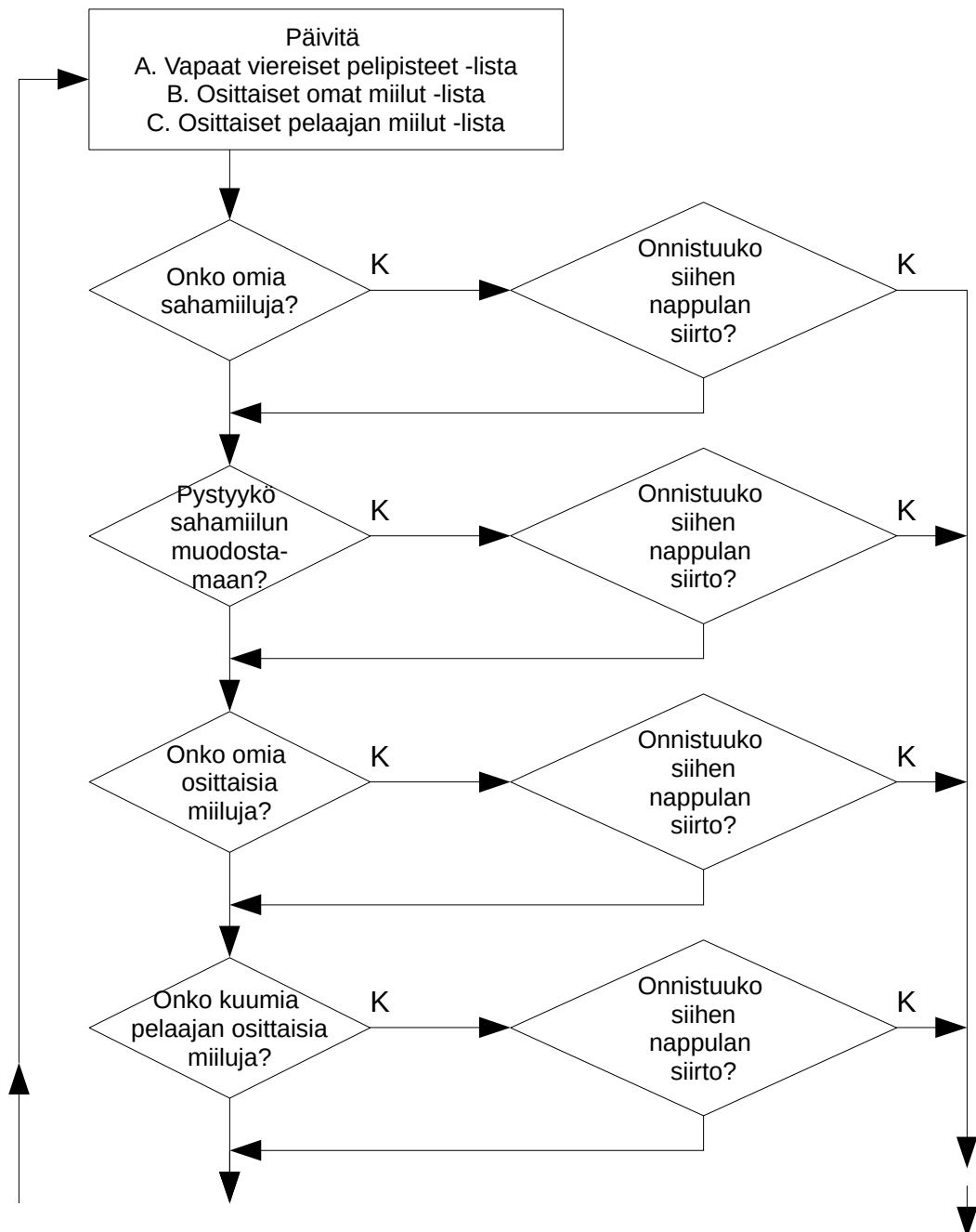
KUVA 20. Puhelimen pelinappulan pelipisteen laskenta.

4.13.2 Pelivaihe

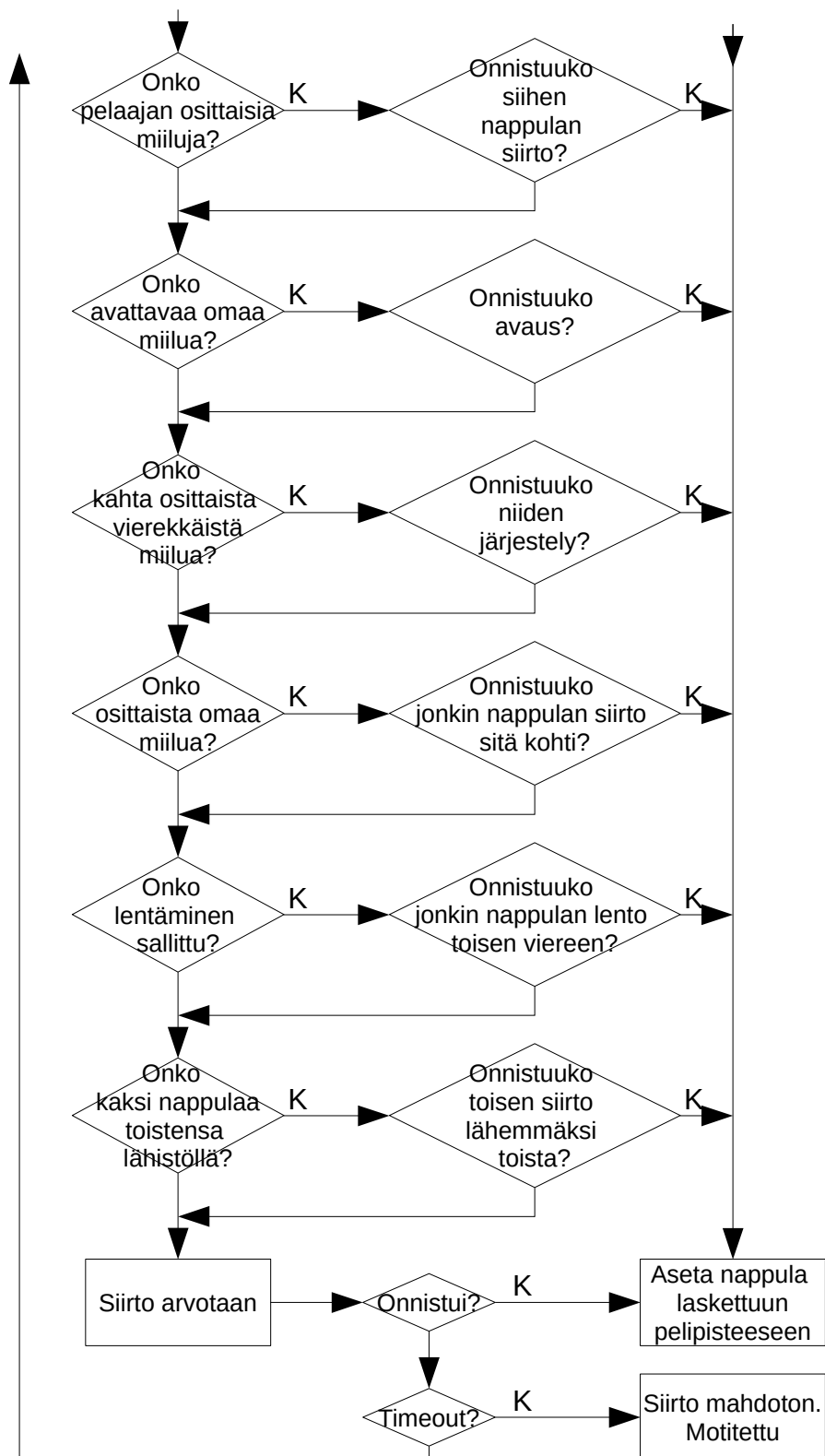
Pelivaiheen aikana puhelimen siirto tehdään kuvissa 21 ja 22 olevan vuokaavion mukaan. Tässä vaiheessa hyödynnetään myös jo edellä mainittuja listoja.

Aluksi etsitään mahdollisia omia miiluja, jotka voitaisiin sulkea, tai ns. sahamiilun muodostamismahdollisuutta. Mikäli sellaisia ei ole, pyritään estämään vastustajan avointen miilujen sulkemismahdollisuus.

Jos edellä mainitut eivät onnistuneet, pyritään avaamaan oma miilu, jos sellainen on. Seuraavana vaihtoehtona on nappuloiden siirto lähemmäksi toisiaan, ja viimeisenä keinona on siirron arvonta. Käytössä on myös silmukkalaskuri, joka päättää laskennan, jos siirtoa ei saada selvitettyä. Tällaisessa tapauksessa pelinappulat ovat motitettuna, ja peli päättyy puhelimen häviöön.



KUVA 21. Puhelimen siirron laskenta (jatkuu seuraavassa kuvassa).



KUVA 22. Puhelimen siirron laskenta (jatkoa).

Ns. kuuma miilu tarkoittaa yllä olevassa vuokaaviossa avoinna olevaa miilua, jonka sulkeminen onnistuu yhdellä siirrolla.

4.13.3 Lopetusvaihe

Kun pelinappuloita on jäljellä enää kolme, sallitaan nappulan lentäminen pelilaudalla mihin tahansa vapaaseen pelipisteeseen. Tämä otetaan huomioon siirron laskennassa kuitenkin käytetään samaa edellä kuvattua päättelymekanismia.

4.14 Pelaajan siirrot

4.14.1 Asetteluvaihe

Pelaajan tullessa siirtovuoroon jäädään odottamaan pelilaudan kosketus -tapahtumaa. Kun sellainen tapahtuu, lasketaan kosketuspistettä lähimmän vapaan pelipisteen paikka ja asetetaan yksi piilossa olevista vapaista pelinappuloista näkyväksi ja sijoitetaan se valittuun pelipisteeseen. Sijoitukseen käytetään *nappulan lento* -tilaa.

4.14.2 Pelivaihe

Pelaajan tullessa siirtovuoroon, jäädään aluksi odottamaan pelinappulan kosketus -tapahtumaa. Kun sellainen tapahtuu, kosketettu pelinappula valikoituu siirron kohteeksi ja sen läpinäkyvyys puolitetaan. Tämän jälkeen jäädään odottamaan pelilaudan kosketus -tapahtumaa. Sen tapahduttua lasketaan kosketuspistettä lähimmän pelipisteen paikka ja sijoitetaan pelinappula siihen. Sijoitukseen käytetään tässäkin *nappulan lento* -tilaa. Esivalittua pelinappulaa on mahdollista vaihtaa vain koskettamalla jotain muuta omista pelinappuloista.

Sääntötarkistus

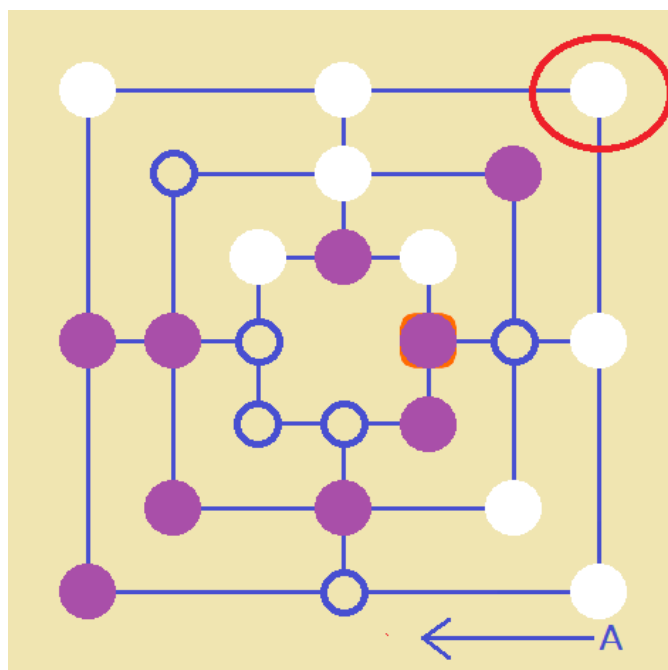
Pelivaiheessa pelinappulan sijoitukseen liittyy aina myös sääntötarkistus. Mikäli aiottu siirto ei ole sääntöjen mukainen, siirtoa ei tehdä ja pelaaja saa kehotuksen yrittää uudelleen. Näitä tapauksia on kahdenlaisia: siirtoyritys pelipisteeseen, joka ei ole nappulan viereinen, tai pelipisteeseen, jossa on jo nappula.

4.14.3 Lopetusvaihe

Kun pelinappuloita on jäljellä enää kolme, sallitaan nappulan lentäminen pelilaudalla mihin tahansa vapaaseen pelipisteeseen. Tämä otetaan huomioon edellisessä luvussa mainituissa sääntötarkistuksessa.

4.15 Miilut

Pelissä miilujen todentamiseen on useampia tapoja. Kaikilla pelipisteellä on muuttuja, johon tallennetaan pelipisteen statustieto (ks. liite 3). Miilun status voi olla joko puhelimen tai pelaajan miilu. Lisäksi pelipisteellä on laskuri, johon lasketaan sen pelipisteen miilujen lukumäärä. Tämä siksi, että tietyn pelipisteen nappula voi olla osana kahdessa eri miilussa. Laskuri on hyödyllinen miilun avauksessa, jolloin miilun pelipisteiden status muuttuu. Tästä on esimerkki kuvassa 23.

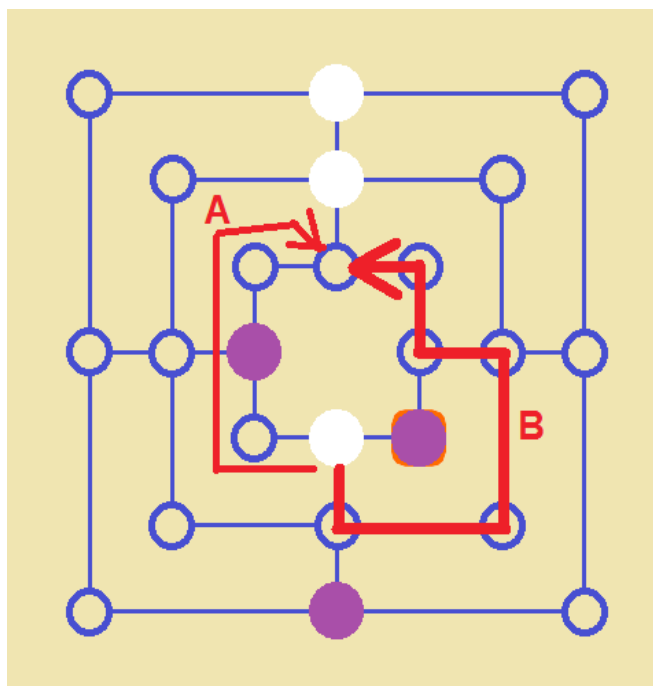


KUVA 23. Miilulaskurin toimintaa. Ympyröidyn nappulan laskurin arvo on kaksi, siirron A jälkeen se on yksi.

Pelilauta-luokassa ylläpidetään myös listaa, jossa ovat kaikki miiluissa olevat pelipisteet. Sitä hyödynnetään aina siirtojen jälkeen miilujen tarkistuksessa. Jos silloin todetaan miilussa oleva mutta listalta puuttuva pelipiste, pelissä on syntynyt uusi miilu ja voidaan aktivoida pelinappulan poistava koodi.

4.16 Navigointi

Joskus puhelimen siirtoja pääteltäessä on tilanne, että pelinappulat ovat hiukan kauempana toisistaan. Tällöin voidaan puhua eräänlaisesta navigoinnista kahden pelipisteen välillä. Tästä on yksinkertaistettu esimerkki kuvassa 24, jossa valkoinen pelaaja miilun muodostaakseen joutuu käyttämään pitempää reittiä B, koska lyhimmillä reitillä A on vastustajan pelinappula esteenä. Joskus mahdollisia reittejä on useita, jolloin niistä on osattava valita järkevin mahdollinen.



KUVA 24. Valkoisen pelinappulan navigoinnin vaihtoehdot miilun muodostamiseksi esimerkkitapauksessa.

Pelissä laskentatapa on kaksivaiheinen. Ensin lasketaan kaikista pelipisteistä kohteeseen tarvittavien siirtojen minimi lukumäärä. Kaikki ne pelipisteet, jotka sisältävät pelinappulan, asetetaan maksimiin. Siirtojen määrä välipisteessä kuvaa tässä tavallaan reitin välipisteiden painoarvoa.

Toisessa vaiheessa virtuaalisesti siirretään haluttua pelinappulaa kohti määrän-päätä seuraten niitä pelipisteitä, joissa painoarvot koko ajan pienenevät. Mikäli tullaan sellaiseen välipisteeseen, josta painoarvo ei enää pienene viereisissä pisteissä, on tultu esteen kohdalle. Tällöin tällaisen välipisteen painoarvoa

kasvatetaan yhdellä ja laskenta aloitetaan alusta. Usein toinen tai kolmaskin kierros päättyy kohti samaa estettä, mutta lopulta kaikki painoarvot tällä reitillä kasvavat riittävän suureksi, jotta lähtötilanteessa valitaan jonkin muu reitti.

Valittua laskentatapaa voidaan pitää eräänlaisena väsytyksen menetelmänä (engl. brute force). Tapa voi olla hiukan raskas, mutta löytää kyllä lyhimmän reitin, mikäli sellainen on olemassa. Toki pelitilanteet ovat hyvin dynaamisia, esimerkiksi valitulle reitille voi vastustaja siirtää oman pelinappulansa esteeksi. Tästä syystä navigoinnissa ei pidetä kirjaa, millä reitillä ollaan vaan se lasketaan aina uudestaan vallitsevan pelitilanteen mukaan. Miinuspuolena on joskus esiintyvä tilanne, jossa pelinappula jää liikkumaan edestakaisin kahden pelipisteen välille.

4.17 Koodin laajuus

Koodirivejä (C#) projektissa on tällä hetkellä noin 2600. Luokkakaavio olisi ollut hyvä lisä liitteeksi, mutta sen automaattista generointia koodista VS ei tukenut.

5 SOVELLUKSEN JULKAISU

5.1 Sovelluksen sertifiointi

Sovelluksen sertifiointivaatimukset on kuvattu Microsoftin www-sivuilla. Tässä luvussa on esitetty niistä oleelliset. (25.)

5.1.1 Sisältö ja sen rajoitukset

Koko sisällön mukaan lukien nimien tulee olla kehittäjän luomia tai hänellä on oltava niihin käyttöoikeus.

Kaupassa julkaistavissa sovelluksissa on kielletty

- laiton toiminta
- liiallinen väkivalta tai sen ihannoitinta
- aseiden käyttö ihmisiä tai eläimiä vastaan
- vihanlietsonta, halventavat, uhkaavat, loukkaavat yms. ihmisoikeuksien vastaiset aiheet
- alkoholi, tupakka, huumeet
- uhkapelit
- liian hävyttömät aiheet.

Myös mahdolliset lakiin perustuvat rajoitukset eri käyttöalueilla tai maissa on otettava huomioon. Sovellukselle on mahdollista antaa ikärajoitus. (25.)

5.1.2 Sovelluspolitiikka

Sovellukselle on määritelty mm. seuraavia vaatimuksia tai ohjeita:

- toimintamalli, mikäli sovelluksen käyttöön liittyy laskutusta
- menettely hyväntekeväisyyden tai vedonlyöntien suhteen
- ilmoitusvelvollisuus mahdollisista sovelluksen käyttöön liittyvistä lisämaksuista
- mahdollisuus estää paikannuksen käyttö
- turvallisuutta ei saa vaarantaa, koskee sekä laitetta että käyttäjää
- yksityisyydensuojan ja tietoturvan säilyttäminen niin laitteessa kuin mahdollisessa datansiirroksakin

- laitteen systeemiasetusten kunnioitus, mm. sovelluksen pitää toimia vaikka ilmoitukset tai tiedonannot olisi estetty
- MPNS (Microsoft Push Notification service) vaatimukset
 - sovelluksen ei tule käyttää ylettömästi datansiirtokapasiteettia
 - ilmoituksia ei saa naamioda
 - ilmoitukset eivät saa olla henkilökohtaisia tai arkaluontoisia
- sovellukseen liittyvä nimi ja muu metadata pitää olla uniikkia
- metadatan pitää myös kuvata sovellusta riittävän tarkasti
- kyvykkyydet on määriteltävä yksiselitteisesti
- lokalisointi on tehtävä kaikille kielille, joita sovellus tukee. (25.)

5.1.3 Asennuspaketti

Asennuspaketin on oltava joko ns. XAP tai AppX-paketti. XAP saadaan suoraan VS:sta. Asennuspaketin tulee

- määritellä sovellukselle nimi (*Title*) ja sen on oltava sama kuin kaupassa
- sisältää oleelliset asennustiedostot kuten **AppManifest.xaml**
- olla kooltaan alle 1 Gt:a. (25.)

5.1.4 Vaatimuksia koodille

Koodi on kehitettävä käyttäen sitä standardia ohjelmointirajapintaa (engl. application programming interface, API), joka sovelluksen käyttöjärjestelmä-versiolle on määritelty. Käännös on tehtävä julkaisu-konfiguraation mukaan eikä siinä saa olla mukana debugaus-tulostuksia yms. (25.)

5.1.5 Lokalisointi

Lokalisointi on tehtävä vähintään yhdelle tuetuista kielistä. Neutraalikielen (WP7) tai oletuskielen (WP8) pitää olla määritelty. (25.)

5.1.6 Ikonit

Sovelluksen asennuspaketissa (XAP) on oltava

- sovellusikoni, 62 x 62 PNG-kuva

- tiili-ikoni, 173 x 173 PNG-kuva.

Lisäksi WP-kauppaa varten tarvitaan seuraavanlaisia ikoneja:

- kaupan katalogin pieni ikoni, 99 x 99 PNG-kuva
- kaupan katalogin iso ikoni, 173 x 173 PNG-kuva
- WP7:lle katalogin ikoni, 200 x 200 PNG-kuva
- WP8:lle katalogin ikoni, 300 x 300 PNG-kuva. (26.)

PNG on häviötön bittikarttagrafiikan tallennusformaatti. (27.)

5.1.7 Esimerkkikuvat sovelluksesta

Tarvitaan vähintään yksi ja maksimissaan kahdeksan kuvankaappausta kutakin tuettua resoluutiota kohden. Kuvat otetaan sovelluksen ollessa käynnissä emulaattorissa ja sen omalla työkalulla eikä niitä saa ottaa oikealla kameralla puhelimesta tai emulaattorista. WP7-sovelluksella käytännössä kuvan resoluutio on aina 480 x 800.

On hyvä huomata, että yleensä emulaattorin oikeassa laidassa olevaa debug-numerokenttää ei saa näkyä esimerkkikuvassa. Sen saa pois näkyvistä editoimalla tiedoston **App.xaml.cs** riviä

Application.Current.Host.Settings.EnableFrameRateCounter = true. Tässä tapauksessa asetus on oltava *false*. (25.)

5.2 Kaupan testaussovellus

Ensimmäinen askel sovelluksen julkaisuun on kokeilla VS:n mukana olevaa testauspakettia (store test kit). Se helpottaa suuresti julkaisun valmistelua, vaikakaan testin osioiden läpäiseminen ei vielä välttämättä takaa hyväksyntää kauppaan. Testifunktiot ovat hiukan riippuvaisia työkaluversiosta, mutta pääpiirteet ovat seuraavanlaisia.

Osa 1

Ensimmäinen osa ohjaa valitsemaan sovellukselle kauppaan tulevat ikonit sekä esimerkkikuvat. Lisäksi tarkistetaan asennuspaketin käännöskonfigurointi (release).

Osa 2

Toinen osa pitää sisällään seuraavat neljä automaattista testiä:

- XAP-asennuspaketin validointi
- kyvykkyyksien validointi
- ikonien validointi
- esimerkkikuvien validointi.

Osa 3

Kolmas osa kattaa neljä monitorointitestiä, jotka on ajettava laitteessa, emulaattori ei kelpaa. Testattavat kohteet ovat sovelluksen

- käynnistysaika
- maksimi muistintarve
- sulkeutumien, (sovellus osaa käsitellä poikkeamat eikä sulkeudu odottamatta)
- Back-näppäimen toiminta.

Osa 4

Viimeinen osa on laajin ja käsittää viisikymmentä erilaista manuaalista testiä.

Niistä tässä on lueteltu muutamia:

- sovelluksen oikea toiminta puhelun tai SMS-viestien saapuessa sekä niiden vastaanoton jälkeen
- Back-näppäimen toiminta kaikilla sivulla
- sovelluksen normaali reagoiminen sen oltua deaktivoitu
- sovelluksen toiminta deaktivoituna (idle-tilan hyödyntäminen)
- sovelluksen ollessa käynnissä, laitteen akku täyteen ladattuna ja näyttö lukittuna, toiminta-aika seuraavaan akun lataukseen on vähintään kuusi tuntia (äänisovellus)
- sovelluksen ollessa käynnissä ja puhelin lentotilassa sekä akku täyteen ladattuna ja näyttö lukittuna, toiminta-aika seuraavaan akun lataukseen on vähintään 120 tuntia (taustasovellus)
- puhelin asetettuna soittamaan musiikkia, sovellus käynnistetään tai suljetaan, musiikin tulee soida normaalisti ilman katkoja (äänisovellus)
- kielivalintojen validointi
- puhelimen väriteemat, toiminta mm. vastakohtaisilla teemoilla.

Osa vaatimuksista on riippuvaisia sovelluksen toiminnasta tai lajista. Esimerkiksi yllä on mainittu muutama ääntä tuottavien tai soittavien äänisovellusten

erityisistä vaatimuksista. Vastaavia löytyy myös videoihin, kuviin tai paikannukseen liittyviin sovelluksiin. Pisin viiden vuorokauden testi vaaditaan taustasovellukselle varmistamaan laitteen toiminta-ajan riittävyys. Taustasovelluksen ominaispiirre on se, että se jää toimimaan taustalle laitteen siirtyessä lukittuun tilaan näyttö sammutettuna (locked screen). (25.)

5.3 Sovelluksen kaupallisuus

Mylly-sovellus on ilmainen, eikä kaupallisia näkökohtia tämän opinnäytetyön yhteydessä tutkittu.

5.4 Kokemukset Mylly-sovelluksen julkaisusta

Käytössäni oli WP8-laite, joten sovelluksen testaukseen laitteessa tarvitsin VS2012:n ja Windows 8:n. Kehitystyön pääosin tein kuitenkin VS2010:n avulla. Testauspaketit näyttävät olevan näissä hiukan erilaisia.

5.4.1 Automaattitestit

VS:n testipaketin avulla käynnistetään sovellus ajoon laitteeseen, ja testattavaa sovellusta käytetään jonkin aikaa normaalisti. Tänä aikana sovelluksen suorituksesta kerätään erilaista tietoa, jota voi analysoida testin jälkeen. Yksi yhteenveto on kuvassa 25.

Osa testeistä esitetään aikajanalla, jolta on mahdollista valita aikaikkuna tarkasteluun. Näistä saatu informaatio ei ole mikään selkein, epäselväksi jäi, kuinka tarkkaan analysointia pitäisi tehdä.

| | | |
|---|------------|--|
| SUMMARY | | |
| 0 Alert(s) | | |
| REPORT | | |
| The different parameters of the app as measured during the analysis session | | |
| Startup time | 0,57 sec | App start time meets requirements. |
| Responsiveness | --- | App is responsive. |
| Total data uploaded | 0,00 KB | Total data uploaded by app is 0,000 KB |
| Total data downloaded | 0,00 KB | Total data downloaded by app is 0,000 KB |
| Battery charge remaining | 8,34 hours | The session consumed 11,00 mAh of battery charge in 220,31 secs. This rate of usage will drain a fully charged standard battery in approximately 8,34 hours |
| Max memory used | 27,46 MB | App max memory usage is 27,46 MB |
| Average memory used | 21,43 MB | App average memory use is 21,43 MB |

KUVA 25. Esimerkki testipaketin antamasta informaatiosta.

5.4.2 Manuaaliset testit

Pelisovellukselle on määritelty mahdollisuus taukoon (pause), johon päästään painamalla Back-näppäintä. Pelin pääsivulle lisäsin tämän ominaisuuden vasta tässä vaiheessa. Aiemmin tämän näppäimen painallus sulki sovelluksen heti.

Vaatimusten mukaisesti lisäsin myös version näyttävään toimintoon ns. tuki-tiedon (engl. support), jonka kautta käyttäjällä on mahdollista ottaa yhteyttä ongelmatilanteessa. Loin sovellukselle ikioman sähköpostiosoitteen (mymills@dnainternet.net), jotta se on suljettavissa mahdollisissa väärinkäyttö-tilanteissa vaikkapa liiallisen roskapostituksen seurauksena.

Aidossa laitteessa tapahtuva testaus on suoritettava vähintään kahdessa eri laitteessa, jotka tukevat sovelluksen näyttöresoluutiota. Tämä tarkoittaa myös sitä, että toinenkin kokeiltava laite on rekisteröitävä kehittäjän laitteeksi, jotta

sovelluksen lataus siihen onnistuu. Kun palaa varsinaisen testauslaitteen pariin, sen rekisteröinti on uusittava.

Erikoisimmasta päästä oleva vaatimus on virus- ja haittaohjelmaskannerin ajo sovelluksen asennuksen jälkeen. En tiedä, haetaanko tällä sitä, ettei sovelluksen asennuspaketissa olisi mitään sellaista, jonka skannerit virheellisesti tulkitsisivat epäilyttäväksi. Tilanne on kuitenkin se, ettei WP:lle oikein ole vaa-
dittuja skannereita, joten vaatimuksen testaus on hankalaa. (25.)

Se, mikä testauspaketissa on toteutettu huonosti, on vaatimusten käsittely. Mm. manuaalisten testien tulos voi olla vain jokin näistä kolmesta:

- hyväksytty (passed)
- hylätty (failed)
- testaamatta (pending).

Koska osa vaatimuksista riippuu sovelluksesta, tarvetta olisi esim. *ei koske tätä sovellusta* -vaihtoehdolle.

5.4.3 Sovelluksen lataus kauppaan

Mikäli kehittäjällä ei ole olemassa olevaa tiliä WP:n kehityssivuilla (dev center), ensimmäinen askel on sen luominen ja mahdollisten kehittäjämaksujen suoritus luottokortilla. Itse rekisteröidyin yksityishenkilönä ja maksoin sen mukaisen maksun, vaikka opiskelijana tuota ei olisi tarvinnut maksaa. Arvelin, että ehkä jatkon kannalta on selvempää, että tiliin liitetty käyttäjätunnus (sähköpostiosoite) ei ole koulun osoite, jonka voimassaolo päättyy pian valmistumisen jälkeen.

Valmistelu

Sovellukselle annetaan aluksi nimi, jonka pitää olla uniikki. Samoin eri kielivaihtoehtoisille annetaan vastaavat nimet. Kauppaan ladataan sovelluksesta käännetty XAP-paketti, ja samassa yhteydessä sovellukselle määritellään

- versio, joka on nelinumeroinen ja esimerkiksi 1.0.0.0
- hinta, ilmaissovellus on luonnollisesti 0,0 €
- sovelluksen laji, kuten peli
- maat, joissa sovellus halutaan julkaistavan
- kieliasetukset, osa tiedosta saadaan asennuspaketista automaattisesti.

5.4.4 Ikäsuositus

Julkaistaessa pelisovellusta tarvitaan tiettyjä alueita ja julkaisumaita varten ikäsuosituksen osoittava sertifiointitodistus. Ellei sellaista ole, voi pelin julkaisu keskeytyä siihen. Tapahtuuko niin oikeasti, ei ole tiedossa. (28.)

Käytännössä tämä vaikuttaa olevan hieman sekava alue, ja yksittäistä kaiken kattavaa todistusta ei ole. Esimerkiksi EU:n sisällä ikäluokitusjärjestelmä PEGI:n (Pan European Game Information) pitäisi olla kelvollinen, näyttäisi kuitenkin, että mm. Saksa ja Portugali voivat vaatia erilliset omat todistuksensa. (29.)

Mylly-sovellukselle hain PEGI:n todistuksen, jonka pitäisi siis kattaa Eurooppa sekä Venäjä. Tuloksena oli K3-luokitus. Samoin hain Yhdysvalloissa ja Kanadassa tarvittavan ESRB:n (Entertainment Software Rating Board) todistuksen. Tuloksena tässä oli *sallittu kaikille*. (30.) Nämä kaksi palvelua ovat maksuttomia ja ne on mahdollista hakea itsepalveluna verkon kautta. Kun vastaa noin kymmeneen kysymykseen, saa sähköpostitse todistuksen heti. Palvelussa vastaan (kyllä tai ei) pelin sisältöä koskeviin kysymyksiin, joista sivusto päättelee sovellukselle järkevän ikäsuosituksen. Esimerkkinä PEGI:n arviointityökalussa olevia kysymyksiä alla. Onko pelissä

- pelottavia elementtejä
- huonoa kielenkäyttöä
- väkivaltaa
- mitään rikoksiin liittyvää
- mitään uhkapeleihin liittyvää
- syrjintää
- huumeita, alkoholia, tupakkaa
- online-yhteys muihin ihmisiin
- seksistisiä elementtejä. (29.)

Sertifiointitodistuksesta on liitteessä 2 esimerkki.

Saksan jätin julkaisumaaksi, vaikka USK-todistusta ei ole. Tämä ei ainakaan ollut este julkaisulle; tosin lautapelit ovat helpoimmasta päästä. (31.)

Maista mm. Brasilia, Kiina ja Taiwan näyttävät olevan vaativimpia ikäsuositusten kannalta. Noiden ohella Mylly-sovellukselle en hakenut julkaisua myöskään seuraavissa maissa: Etelä-Afrikka, Etelä-Korea, Japani, Australia ja Uusi-Seelanti. Jos jokin sertifiointi on jo saatu, se helpottanee hakuprosessia em. maihin. Esimerkiksi Brazilian todistuksen voinee hakea online-työkalulla ole-massa olevan PEGI tai ESRB:n avulla. (28.)

5.4.5 Mainostekstit

Kuten jo aiemmin mainittiin, julkaisua varten tulee olla käynnissä olevasta sovel-luksesta kuvankaappauksia. Näiden tarkoitus on olla eräänlainen mainos kau-passa. Kuitenkin on hyvä huomata, että eri kielivaihtoehtojen tarvi-taan omat mainoskuvansa.

Sama koskee myös tekstimuotoisia kuvauksia. Näihin on syytä varautua etukä-teen, jotta haluttu teksti olisi mahdollisimman helppo liittää latausprosessin ai-kana. Omalla kohdallani jouduin miettimään mainostekstit turhan nopeasti, jol-loin niistä tuli ehkä liian suppeita sekä vieraskielisistä teksteistä ei tullut parhaita mahdollisia.

5.4.6 Palaute

WP-kaupasta en saanut palautetta julkaisuprosessin aikana ja sovellus serti-fioitiin nopeasti, alle tunnissa. Se oli pienoinen yllätys, sillä sertifiointi voi kestää useita päiviä. Kauppaan se ilmestyi yleisesti saataville noin kuusi tuntia tämän jälkeen. Sähköpostitse saatu sertifiointi-ilmoitus on kuvassa 26.

5.4.7 Tilastotietoa

Kehityssivujen kautta voi seurata mm. latausten määrää. Ensimmäisen kuu-kauden aikana Mylly-sovellusta oli ladattu noin 800 kertaa, joista suurin osa Suomessa. Tilaston saa rajattua maan ohella myös kielen mukaan. Mylly-sovellusta englannin tai ruotsin kielellä oli tänä aikana ladattu noin kymmenen kertaa.

Myös sovelluksen kaatumisia seurataan. Tältä osin pelini vakaus näyttää hyvältä, koska seuranta-ajalla ei kaatumisia näytä olleen lainkaan. Arvosteluja kauppaan oli jätetty yhdeksän kappaletta ja niiden perustella laskettu keskiarvo sovellukselleni on noin kaksi tähteä.

Congratulations! myMills app has successfully passed certification to be included in the Windows Phone Store.

You can now publish your app to the Windows Phone Store.

If you chose to automatically publish your app during the submission process, it will be published within a few hours. If you didn't choose to automatically publish your app, you can do so now by going to your app list at <https://dev.windowsphone.com/fi-FI/applicationlist>.

You'll need your Microsoft account to access your account on Dev Center.

Thank you,

The Windows Phone Team

Microsoft respects your privacy. To learn more, read [Privacy and Cookies](#).

Microsoft Corporation | One Microsoft Way | Redmond, WA 98052

To ensure reliable delivery of future communications from the Windows Phone team, please add mktpdev@microsoft.com to your safe senders list.

Microsoft

KUVA 26. Projektin ensimmäisen vaiheen päätös, sovellus on julkaistu.

6 POHDINTA

Työssä tavoitteena oli älypuhelinsovelluksen kehitystyö sekä sen julkaisu sovel-luskauppaan yleisesti ladattavaksi. Pääpaino oli osin enemmän kehitystyön tar-kastelussa kuin varsinaisessa sovelluksessa.

WP-sovelluksen kehitystyö on kohtuullisen suoraviivaista. Käyttöliittymästä saadaan selkeä, visuaalinen ja voisi jopa sanoa oikein hieno, mikäli ulkoasun suunnitteluun halutaan käyttää aikaa. Tarjolla olevat työkalut toimivat hyvin ja ilman suurempia ongelmia. Oikeastaan jopa erinomaisesti, jos verrataan Android-leirin vastaaviin SDK:ihin.

Enemmän haasteita on tarjolla perusohjelmoinnissa. Tapahtumakeskeinen toi-mintamalli voi vaatia ainakin aluksi pohdintaa, miten koodin toimivuus saadaan aikaiseksi. Sama pätee luokkiin ja olioihin, mikäli ne eivät ole täysin tuttuja ke-hittäjälle. Toisaalta kehitystyön aikana varmasti huomataan, kuinka ne ovat erit-täin tehokkaita ja helpottavat koodaamista suuresti.

Kehitystyössä tuli selvästi ilmi se ohjelmoinnin yleinen pullonkaula, että välillä koodia syntyy hyvinkin hitaasti. Tai kyllä koodirivejä aikaansaadaan, mutta halu-tun ominaisuuden toimivaksi hiominen vaatii aikansa ja virheenkorjauksia. Vaik-ka Mylly-sovellus näyttää yksinkertaiselta, sen toiminnassa on yllättävän paljon huomioitavaa. Etenkin pelisovelluksen dynaamisuus on tietynlainen haaste, koska erilaisia tilanteita on tuhansia. Näin ollen tietyn koodinpätkän on parempi taipua käsittelemään mahdollisimman monta samankaltaista pelitilannetta.

Jatkokehitettävääkin vielä jäi. Alun perin oli tavoitteena kaksi tai kolmekin peli-tasoa, mutta aikataulusyiden vuoksi ne olivat unohdettava. Totuuden nimissä on todettava, että todennäköisesti vaikean pelitason kehitystyö voi vastata nyt ai-kaansaadun koko sovelluksen kehitystä. Puhumattakaan, että saataisiin kehi-tettyä vaikkapa voittamaton, aina tasapeliin päättyvä peliäly Ralph Casserin ideoiden pohjalta (32, s. 12). Käytännössä noin vaativa ja raskas päättelykoodi kannattanee jo sijoittaa serverille, jolle laite lähettää pelitilanteen pääteltäväksi ja saa takaisin ohjeen seuraavan siirron tekemiseksi. Toki tällöin miinuksena on, että tarvitaan käytännössä jatkuva yhteys verkkoon, jolloin rajoitetaan jossain määrin käyttötilanteita. Esimerkiksi lentokoneessa voi nykyisin käyttää koko

lennon ajan laitetta, jonka lentotila on aktivoitu. Siispä ainakin serverin tuki olisi silloin sovelluksen saavuttamattomissa.

Pienistä puutteista huolimatta itse olen tyytyväinen nyt aikaan saatuun aloittelijoille sopivaan helppoon pelitasoon. Visuaalisuudesta voisi saada vielä enemmän irti. Esimerkiksi lisäämällä varjostuksia pelinappuloihin ne saisi lähestymään 3D:n kaltaisia ominaisuuksia.

LÄHTEET

1. Tilastokeskus. 2013. Väestön tieto- ja viestintätekniikan käyttö. Saatavissa: http://www.stat.fi/til/sutivi/2013/sutivi_2013_2013-11-07_tie_001_fi.html. Hakupäivä 1.9.2014.
2. Windows Phone. 2014. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Windows_Phone. Hakupäivä 12.9.2014.
3. GSM Arena. 2014. Saatavissa: <http://www.gsmarena.com/>. Hakupäivä 1.9.2014.
4. Brix, Todd. 2014. New Dev Center lifetime registration and benefits program. Saatavissa: <http://blogs.windows.com/buildingapps/2014/09/17/new-dev-center-lifetime-registration-benefits-program/>. Hakupäivä 2.11.2014.
5. Microsoft DreamSpark. 2014. Saatavissa: <https://www.dreamspark.com/Student/Windows-Store-Access.aspx>. Hakupäivä 2.11.2014.
6. Visual Studio. 2014. Saatavissa: <http://www.visualstudio.com/>. Hakupäivä 15.8.2014.
7. Microsoft Silverlight. 2014. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Microsoft_Silverlight. Hakupäivä 15.10.2014.
8. Microsoft XNA. 2014. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Microsoft_XNA. Hakupäivä 15.10.2014.
9. Windows Phone Emulator for Windows Phone 8. 2014. Saatavissa: <http://msdn.microsoft.com/en-us/library/windows/apps/ff402563%28v=vs.105%29.aspx>. Hakupäivä 15.11.2014.

10. Zune software for your PC. 2014. Saatavissa:
<http://www.windowsphone.com/en-US/How-to/wp7/zune-software>.
Hakupäivä 15.11.2014.
11. C Sharp. 2014. Wikipedia. Saatavissa:
http://fi.wikipedia.org/wiki/C_sharp. Hakupäivä 15.10.2014.
12. Common Language Runtime. 2013. Wikipedia. Saatavissa:
http://en.wikipedia.org/wiki/Common_Language_Runtime.
Hakupäivä 16.10.2014.
13. App platform compatibility for Windows Phone 8. 2014. Saatavissa:
<http://msdn.microsoft.com/en-us/library/windows/apps/jj206947%28v=vs.105%29.aspx>.
Hakupäivä 4.10.2014.
14. Windows Phone Versions. 2014. Saatavissa:
<http://msdn.microsoft.com/library/windows/apps/hh202996%28v=vs.105%29.aspx>. Hakupäivä 4.10.2014.
15. The three buttons: Back, Start, and Search. 2014. Saatavissa:
<http://www.windowsphone.com/en-us/how-to/wp8/phones-and-hardware/the-three-buttons-back-start-and-search>. Hakupäivä 20.11.2014.
16. Itzkovitch, Avi. 2012. Designing For Device Orientation: From Portrait To Landscape. Smashing Magazine 2012/August issue. Saatavissa:
<http://www.smashingmagazine.com/2012/08/10/designing-device-orientation-portrait-landscape/>. Hakupäivä 5.9.2014.
17. Globalization and localization for Windows Phone 8. 2014. Saatavissa:
<http://msdn.microsoft.com/library/windows/apps/ff637522%28v=vs.105%29.aspx>. Hakupäivä 1.10.2014.
18. How to build a localized app for Windows Phone 8. 2014. Saatavissa:
<http://msdn.microsoft.com/en->

[us/library/windows/apps/ff637520%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff637520%28v=vs.105%29.aspx).

Hakupäivä 4.10.2014.

19. How to localize an app title for Windows Phone 8. 2014. Saatavissa:

[http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/ff967550%28v=vs.105%29.aspx)

[US/library/windows/apps/ff967550%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff967550%28v=vs.105%29.aspx).

Hakupäivä 4.10.2014.

20. How to determine app capabilities for Windows Phone 8. 2014.

Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/gg180730%28v=vs.105%29.aspx)

[us/library/windows/apps/gg180730%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/gg180730%28v=vs.105%29.aspx).

Hakupäivä 17.10.2014.

21. How to modify the app manifest file for Windows Phone 8. 2014.

Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/jj247542%28v=vs.105%29.aspx)

[us/library/windows/apps/jj247542%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/jj247542%28v=vs.105%29.aspx).

Hakupäivä 17.10.2014.

22. How to create a splash screen for Windows Phone. 2014.

Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/ff769511%28v=vs.105%29.aspx)

[us/library/windows/apps/ff769511%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff769511%28v=vs.105%29.aspx).

Hakupäivä 25.10.2014.

23. Dispatcher Class. 2014. Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/system.windows.threading.dispatcher%28v=vs.105%29.aspx)

[us/library/windows/apps/system.windows.threading.dispatcher](http://msdn.microsoft.com/en-us/library/windows/apps/system.windows.threading.dispatcher%28v=vs.105%29.aspx)

[%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/system.windows.threading.dispatcher%28v=vs.105%29.aspx). Hakupäivä 25.10.2014.

24. How to use a background worker for Windows Phone 8. 2014.

Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/cc221403%28v=vs.105%29.aspx)

[us/library/windows/apps/cc221403%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/cc221403%28v=vs.105%29.aspx).

Hakupäivä 1.9.2014.

25. App certification requirements for Windows Phone. 2014.

Saatavissa: [http://msdn.microsoft.com/en-](http://msdn.microsoft.com/en-us/library/windows/apps/hh184843%28v=vs.105%29.aspx)

[us/library/windows/apps/hh184843%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windows/apps/hh184843%28v=vs.105%29.aspx).

Hakupäivä 23.10.2014.

26. Engle, Rick. 2014. Creating Windows Phone 7 Application and Marketplace Icons. Saatavissa: <http://msdn.microsoft.com/en-us/expression/gg317447.aspx>. Hakupäivä 1.10.2014.
27. PNG. 2013. Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/PNG>. Hakupäivä 19.11.2014.
28. Game Ratings. 2014. Saatavissa: <http://msdn.microsoft.com/en-us/library/windows/apps/jj206734%28v=vs.105%29.aspx>. Hakupäivä 2.11.2014.
29. Pan European Game Information. Saatavissa: <http://www.pegi.info/fi/>. Hakupäivä 2.11.2014.
30. Entertainment Software Rating Board. Saatavissa: <http://www.esrb.org/index-js.jsp>. Hakupäivä 2.11.2014.
31. Welcome to the USK. Saatavissa: <http://www.usk.de/en/>. Hakupäivä 2.11.2014.
32. Casser, Ralph. 1996. Solving Nine Men's Morris. Games of No Chance, MSRI Publications. Saatavissa: <http://library.msri.org/books/Book29/files/gasser.pdf>. Hakupäivä 1.8.2014.

LIITTEET

Liite 1 KOODIESIMERKKI

Liite 2 IKÄSUOSITUSSERTIFIKAATTI

Liite 3 ESIMERKKEJÄ LUOKISTA

Esimerkkinä on yksi pelipisteen siirroksissa käytetty laskentatapa.

```
// Method which returns next spot
public int nextSpot(int spot)
{
    spot++;
    if( spot % BOX_OFFSET == 0 )           // BOX_OFFSET = 8
        spot -= BOX_OFFSET;
    return spot;
}
// Method which returns previous spot
public int prevSpot(int spot)
{
    if (spot % BOX_OFFSET == 0)
        spot += BOX_OFFSET;
    spot--;
    return spot;
}
// Method which returns next spot upwards, handle negative in calling code
public int upSpot(int spot)
{
    spot -= BOX_OFFSET;
    return spot;
}
// Method which returns next spot downwards, handle over table size in
// calling code
public int downSpot(int spot)
{
    spot += BOX_OFFSET;
    return spot;
}
```

Esimerkkinä alla on Piece-luokan konstruktori.

```
public Piece(Grid canvas)
{
    pieceImg = new Image();
    canvas.Children.Add(pieceImg);
    currentLocation = new Position();
    lastLocation = new Position();
    destination = new Position();
    delta = new Position();
    freeAdjacentSpots = new List<int>();
    flyHops = 0;
    pieceImg.HorizontalAlignment = HorizontalAlignment.Left;
    pieceImg.VerticalAlignment = VerticalAlignment.Top;
    isLocked = true;
    spotNumber = -1;
    alive = false;

    pieceImg.Tap += new EventHandler<GestureEventArgs>(pieceImg_Tap);
}
```



PEGI Age Rating License

Name of the Company: Jukka Raappana

Name of Game: myMills

Registered Number: 42627

Platform: Windows Phone

PEGI URN: 60WW3W-P71N00-003000-000000

We confirm receipt of the registration of the above-mentioned interactive software myMills in the name of your company Jukka Raappana on the Windows Phone platform with the age rating logo 3 and content descriptor(s) None described hereunder and as prescribed by the PEGI game rating system.

In consideration of the above we hereby grant Jukka Raappana and its subsidiaries the non-exclusive right to use the above mentioned Logo and Descriptor(s) in all PEGI territories for the duration of its commercial life and in accordance with the Labelling Guidelines as made available by PEGI S.A. on the understanding that the content of the Interactive Software shall not be subject to any alteration that might cause a change to its designated age rating.

We also grant Jukka Raappana and its subsidiaries the non-exclusive right to:

- Represent and to reproduce by any means in all PEGI Territories the Logo and Descriptor(s), in combination with the name and/or any picture representing the Interactive Software.
- Sub-license the use of the Logo and Descriptor(s) in combination with the name and/or any picture representing the Interactive Software in all PEGI Territories.

These rights are granted for the sole purpose of marketing and advertising of the Interactive Software on the Windows Phone platform. If the Interactive Software is to be released on another platform, it must be subject of a separate age rating request.

Yours faithfully,

PEGI S.A. & Interactive Software Federation of Europe



TAULUKKO 8. Piece-luokan attribuutit.

| Attribuutti | Tyyppi | Kuvaus |
|-------------------|-----------|---|
| arrayNumber | int | Indeksinumero nappulataulukossa, jonka koko on yhdeksän per pelaaja |
| spotNumber | int | Pelipisteen numero pelilaudalla |
| isLocked | Boolean | Lippu, onko nappula lukittu. Esim. liikuttelu on kielletty |
| currentLocation | Position* | Nykyinen koordinaatti näytöllä |
| destination | Position | Nappulan lentäessä kohteen koordinaatti |
| delta | Position | Nappulan lentäessä koordinaattien muutosarvo |
| flyHops | int | Nappulan lentäessä jäljellä olevien hyppyjen lukumäärä |
| imgSrc | string | Nappulan kuvan tiedostonimi |
| pieceImg | Image | Nappulan kuva-image |
| aLive | Boolean | Lippu, onko nappula pelilaudalla vai poistettu |
| freeAdjacentSpots | List<int> | Lista, johon talletetaan viereiset vapaana olevat pelipisteet |

*Position on yksinkertainen koordinaattiluokka (double X, double Y)

TAULUKKO 9. Piece-luokan metodit.

| Metodi | Tyyppi | Kuvaus |
|---|--------------|--|
| Piece(Grid canvas) | void | Konstruktori, varaa tarvittavan muistin olioille ja asettaa attribuutit oletustilaan |
| getXYbySpot(int spotNbr) | Position | Palauttaa koordinaatit, kun tiedetään pelipiste |
| xyToSpot(Position tappedXY) | int | Palauttaa pelipisteen, kun tiedetään koordinaatit |
| setPosition(Position position) | Boolean | Asettaa kuva-imagen haluttuun koordinaatiston pisteeseen |
| fly() | Boolean | Kun päätepiste on asetettu, lennättää pelinappulaa ruudulla kohteeseen |
| setFlyDestination(Position position) | void | Asettaa pelipisteen päämäärän |
| setMargin(Position newPos) | void | Asettaa kuva-imagen marginaalin eli koordinaatit |
| setOpacity(double newOpacity) | void | Asettaa kuva-imagen näkyvyyden halutuksi |
| setImage(string str) | Boolean | Asettaa kuva-imagen tiedostonimen |
| PieceMovementDone(PieceEventArgs e) | Virtual void | Tapahtumankäsittelijä, jota kutsutaan kun liikkumistoiminnot on saatu päätökseen |
| pieceImg_Tap(object sender, GestureEventArgs e) | void | Tapahtumankäsittelijä, jota kutsutaan, kun kuva-imagea on näpätetty |

TAULUKKO 10. PieceSpot-luokan attribuutit.

| Attribuutti | Tyyppi | Kuvaus |
|-------------|---------|--|
| piece | Piece | Tällä viitataan siihen Piece-olioon, joka on asetettu tähän pelipisteeseen |
| isEmpty | Boolean | Lippu, joka kertoo onko pelipiste tyhjä vai ei |
| spotStatus | Status* | Pelipisteen tila yleisemmin |
| millCtr | byte | Lasketaan monessako miilussa pelipisteen nappula on |

*Status on enum-tyyppin muuttuja ja pitää sisällään tilat: tyhjä, puhelimen miilu, sekä pelaajan miilu

TAULUKKO 11. GameBoard-luokan attribuutit.

| Attribuutti | Tyyppi | Kuvaus |
|----------------|-------------|---|
| boardTable | PieceSpot[] | Taulukko, joka kuvastaa pelilaudan pistetä |
| boardTableUndo | PieceSpot[] | Vastaava taulukko "undo" toiminnallisuutta varten (ei toteutettu) |
| millList | List<int> | Lista, johon talletetaan aktiiviset miilut |
| naviList | List<int> | Lista, johon talletetaan siirtymäreitti pisteestä A → B |

TAULUKKO 12. GameBoard-luokan metodit.

| Metodi | Tyyppi | Kuvaus |
|---|-------------|--|
| initBoardTable() | void | Alustaa pelipistetaulukon |
| ClearBoardTableSpot(int spot, Boolean delAll) | void | Nollaa taulukon tietyn pisteen tiedot, tarvittaessa nollaa pelinappulan pois näkyvistä |
| isMoveValid(int fromSpot, int toSpot) | Boolean | Testaa, onko siirto sääntöjen mukainen |
| getBoardTable() | PieceSpot[] | Palauttaa pelipistetaulukon osoitteen |
| checkForMill() | Status | Tarkistaa miilujen tilanteen |
| isNewMill(int spot) | Boolean | Tarkistaa, onko uusia miiluja |
| isInMillTable(int spot) | Boolean | Tarkistaa, onko tietty pelipiste jo miilulistassa |
| dropFromMillTable(int spot) | Boolean | Pudottaa pelipisteen pois listalta |
| isSpotInMill(int spot) | Boolean | Tarkistaa, onko tietty pelipiste miilussa |
| checkBoxMill(int spot) | Boolean | Tarkistaa neliön mukaiset miilut |
| checkCrossMill(int spot) | Boolean | Tarkistaa ristin mukaiset miilut |
| decMillCtr(int spot) | void | Vähentää miilulaskuria |
| incMillCtr(int spot) | void | Lisää miilulaskuria |
| prevSpot(int spot) | int | Palauttaa edellisen pelipisteen |
| nextSpot(int spot) | int | Palauttaa seuraavan pelipisteen |
| upSpot(int spot) | int | Palauttaa ylemmän pelipisteen |
| downSpot(int spot) | int | Palauttaa alemman pelipisteen |
| possibleMove(int spot) | int | Palauttaa tiedon, voiko tiettyyn pelipisteeseen siirtyä |
| seekPartialMills(List<int> openMills, Status whosPiece) | int | Etsii osittaisia ts. kahden nappulan omaavia mahdollisia miiluja |

| | | |
|--|----------------------|--|
| <code>seekPartialMillsEmptyPiece(List<int> openMills, int spot)</code> | <code>int</code> | Etsii osittaisia ts. kahden nappulan omaavia mahdollisia miiluja, palauttaa tyhjän pelipisteen |
| <code>isPartialMillClosable(List< int> openMills, int toSpot, Piece[] piece, Boolean returnEmpty)</code> | <code>int</code> | Tutkii, onko jokin osittainen miilu mahdollista sulkea yhdellä siirrolla |
| <code>seekDoublePartialMillPieces (List<int> openMills, Boolean returnMovable)</code> | <code>int</code> | Tutkii, onko jokin osittaisen miilun nappulat tuplassa osittaisessa miilussa |
| <code>testMove(int fromSpot, int toSpot)</code> | <code>Boolean</code> | Testaa siirron, tuleeeko miilua |
| <code>testFlyMove(int fromSpot, int toSpot)</code> | <code>Boolean</code> | Testaa siirron, tuleeeko miilua lentämällä |
| <code>distance(int fromSpot, int toSpot)</code> | <code>int</code> | Laskee kahden pelipisteen lyhyimmän etäisyyden |
| <code>navigateToSpot(int fromSpot, int toSpot, Boolean returnDistance)</code> | <code>int</code> | Etsii mahdollisen lyhimmän reitin kahden pelipisteen välillä |
| | | |